HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Electronics, Communication and Automation
Department of Signal Processing and Acoustics

Tapani Pihlajamäki

Multi-resolution Short-time Fourier Transform Implementation of Directional Audio Coding

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo, August 10, 2009

Supervisor: Matti Karjalainen
Instructors: Ville Pulkki

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author:	Tapani Pihlajamäki
Name of the thesis:	Multi-resolution Short-time Fourier Transform Implementation of Directional Audio Coding
Date:	August 10, 2009 **Number of pages:** 79 + vii
Faculty:	Electronics, Communication and Automation
Professorship:	S-89
Supervisor:	Prof. Matti Karjalainen
Instructors:	Docent Ville Pulkki

The study of spatial hearing has been a prominent topic in the acoustical community. Research has also produced new ways for spatial audio reproduction. Recently proposed Directional Audio Coding is one of them. It is a method for processing and reproducing spatial audio. This is done with relatively simple algorithms analyzing the direction of arrival and the diffuseness from a sound signal in B-format form. This analyzed information is then used to synthesize direct sound and diffuse sound separately to produce a representation of the original soundfield which sounds similar to the human listener when compared to the original. These algorithms are based on a few psychoacoustical assumptions.

In this thesis, in addition to evaluation of basic algorithms, a few new methods are proposed. These are: the application of multi-resolution short-time Fourier transform, frequency bin based processing, a hybrid decorrelation method and a time varying phase modulating decorrelation method.

In informal evaluation, it was found that the use of multiple resolutions increases the quality of sound. Bin based processing, however, did not increase subjective quality. Also, new decorrelation methods did not produce any enhancement compared to the previously established methods. Also, these results were achieved with a great cost in calculation needs and use of alternative methods are recommended for all but the multi-resolution case.

As a partial task for this thesis, a software library of Directional Audio Coding was developed. This library enables easy portability and application of Directional Audio Coding method in multitude of situations with a highly parametrized control over the performance.

Keywords: Abstract

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Tapani Pihlajamäki
Työn nimi:	Directional Audio Coding -menetelmän toteutus käyttäen monitarkkuuksista lyhytaikaista Fourier-muunnosta
Päivämäärä:	10.8.2009 **Sivuja:** 79 + vii
Tiedekunta:	Elektroniikka, tietoliikenne ja automaatio
Professuuri:	S-89
Työn valvoja:	Prof. Matti Karjalainen
Työn ohjaajat:	Dosentti Ville Pulkki

Tilakuulon tutkimus on ollut tärkeä aihe akustiikan alalla. Tutkimus on myös tuottanut tilaäänen toistoon uusia keinoja. Äskettäin esitetty Directional Audio Coding (DirAC) -menetelmä on yksi tuloksista. Se on tarkoitettu tilaäänen prosessointiin ja toistoon. Tämä saavutetaan suhteellisen yksinkertaisella algoritmilla, joka määrittää B-formaatti muodossa olevasta äänisignaalista äänen tulosuunnan ja diffuusisuuden. Käyttämällä näitä tietoja voidaan ei-diffuusi ääni ja diffuusi ääni syntetisoida erikseen ja toistaa alkuperäinen äänikentän niin, että se kuulostaa ihmiselle samalta. Nämä algoritmit perustuvat muutamaan psykoakustiseen oletukseen.

Tässä työssä ehdotetaan muutamaa uutta menetelmää sekä arvioidaan vanhoja toteutuksia. Nämä ovat: monitarkkuuksinen lyhytaikainen Fourier-muunnos, laskenta perustuen diskreetteihin taajuusyksiköihin, yhdistelmämenetelmä dekorrelaatioon ja aikamuuttuvasti vaihetta moduloiva dekorrelaatiomenetelmä.

Epäformaaleissa testeissä huomattiin, että useamman tarkkuuden käyttäminen paransi äänenlaatua. Diskreettien taajuusyksiköiden avulla laskeminen sen sijaan ei tuottanut havaittavaa etua. Samoin uudet dekorrelaatiomenetelmät eivät parantaneet tulosta aiempiin menetelmiin verrattuna. Lisäksi uusien ominaisuuksien lisääminen lisäsi algoritmin laskennallista vaativuutta merkittävästi. Tästä johtuen vaihtoehtoisia menetelmiä suositellaan kaikissa muissa tapauksissa paitsi monitarkkuusmenetelmän tapauksessa.

Osana diplomityötä toteutettiin myös ohjelmakirjasto Directional Audio Coding -menetelmän käyttöön. Tämä kirjasto mahdollistaa menetelmän muuntamisen ja soveltamisen useisiin tilanteisiin ja tarjoaa paljon säätöjä menetelmän toiminnan muokkaamiseen.

Avainsanat: Tiivistelmä

Acknowledgements

First and foremost I want to thank my instuctor, Docent Ville Pulkki, who offered me the task for this thesis work and supported by providing new ideas and correcting my misunderstandings. Secondly, my thanks go to the supervisor of my thesis, professor Matti Karjalainen. Although he did not participate that much during my thesis work, it was still comforting to know that his support was there and at the final steps of my thesis work, his help was invaluable.

I also want to thank my co-workers Jukka Ahonen, Mikko-Ville Laitinen, Juha Vilkamo, Marko Hiipakka, Marko Takanen and Olli Santala. They provided a lot of friendly support during my work and discussed ideas related to the thesis work. Especially Ahonen, Laitinen and Vilkamo were helpful for their knowledge in Directional Audio Coding.

My gratitude goes also to everyone else in the Department of Signal Processing and Acoustics in Helsinki University of Technology. Friendly discussion with them was one of the things keeping me sane during the process.

Otaniemi, August 10, 2009

Tapani Pihlajamäki

Contents

Abbreviations

DFT	Discrete Fourier transform
DirAC	Directional Audio Coding
ERB	Equivalent rectangular bandwidth
FFT	Fast Fourier transform
FIR	Finite impulse response
IACC	Inter-aural cross-correlation
IDFT	Inverse discrete Fourier transform
IFFT	Inverse fast Fourier transform
IIR	Infinite impulse response
ILD	Inter-aural level difference
ITD	Inter-aural time difference
MRSTFT	Multi-resolution short-time Fourier transform
STFT	Short-time Fourier transform
VBAP	Vector base amplitude panning

Chapter 1

Introduction

It has been roughly 132 years since Thomas Alva Edison invented the first sound reproduction mechanism, the phonograph. Through time the technology has evolved significantly and brought new wonders also in audio technology. Gramophone records and two-channel stereophonic transmissions were concurrently introduced and developed through the beginning of 20th century. When Philips introduced Compact Disc at 1979, the dawn of digital form broke. Nowadays, the most common form is digital data in "mp3" files. However, this evolution has happened only for the storage media.

Actual sound reproduction paradigm has not changed that much. The most common home system tends to be a medium-sized stereo system offering passable quality. In most cases, the quality of the media is much better than the system can produce. However, during last ten years, multichannel systems have finally started to become more popular at homes, thanks to DVD-movies. However, there are still purists who say that only monaural systems are "pure".

Still, there is room for development as the sound reproduction is not yet perfect. One can verify this by going to a music concert and noticing that the experience is on a completely different level compared to any recording. Current multichannel concert recordings come close but do not capture all nuances of the performance. This, however, can be enhanced with the newest technologies and this thesis will focus on one of them called Directional Audio Coding (DirAC). It is a relatively new method proposed only a few years ago but has already received some interest. The groundwork for its algorithms was already created for Spatial Impulse Response Rendering technology.

The aim of this thesis is to produce a high quality short-time Fourier transform based version of DirAC. As there has been already few implementations of DirAC, the prime solution is to test and apply new algorithms. Also, the aim is to produce a software library which can be easily used in further development of DirAC.

This thesis contains ten chapters with this introduction being the first one. Chapters 2–5 will contain the background knowledge in the areas of physics of sound, hearing and psychoacoustics, signal processing and sound reproduction needed in this thesis. After that, chapters 6–8 describe the DirAC algorithm, its implementation in this thesis and the results. Chapter 9 is dedicated for describing the produced software library and chapter 10 concludes the thesis.

Chapter 2

Physics of Sound

This chapter dwells in the properties of sound related to physics. First, fundamental properties are studied and then the propagation sound through air is described.

2.1 Fundamentals

Elementary physics state that sound is a movement of particles and changes of pressure in the air. Actually, air is not the only possible conduit. More precisely sound can be defined as longitudinal pressure waves moving through any medium (Rossing et al., 2002). These pressure waves can then be perceived as sound if they arrive to the human ear.

Sound waves have two elementary variables which can be used to define the properties of the wave. They are sound pressure p and particle velocity $\vec{u}$. Sound pressure is the same thing as normal pressure in any fluid but particle velocity might not be that clear. It is not the speed at which the sound wave moves through space but the speed at which wave transmitting particles move in space. Notable is that although particle velocity alternates like the sound pressure, particle velocity also has a direction.

There are also a few important constants which depend on the medium through which sound is traveling. These constants are the speed of sound c, the mean density of the medium ρ_0 and the characteristic acoustic impedance of the medium Z_0. The medium is usually air at room temperature (20° Celsius). In the context of this thesis, this is also assumed and thus the constants have following values: $c = 343.2\frac{m}{s}$, $\rho_0 = 1.204\frac{kg}{m^3}$ and $Z_0 = 413.2\frac{Ns}{m^3}$.

With these definitions it is possible to define other useful variables. They are sound intensity $\vec{I}$, sound energy E and diffuseness ψ. Intensity is defined as the product of sound pressure and particle velocity (Eq. 2.1) (Fahy, 1989). It describes the net flow of energy through space.

$$\vec{I} = p\vec{u} \tag{2.1}$$

Sound energy also depends on the pressure and particle velocity. It is defined with equation

$$E = \frac{1}{2}\rho_0 \left(\frac{p^2}{Z_0^2} + \|\vec{u}\| \right) \tag{2.2}$$

(Fahy, 1989) and tells the energy density the sound wave has.

Diffuseness is useful to explain well in context of this thesis. Formally it is defined that in a perfectly diffuse field, the energy density is constant in all points within the volume of interest (Nélisse and Nicolas, 1997). What this really means is that with a perfectly diffuse sound, there is no energy transmission and thus no clear direction for the sound wave. Diffuseness can be calculated from the active intensity and energy with equation (Merimaa and Pulkki, 2005)

$$\psi = 1 - \frac{\|\langle \vec{I} \rangle\|}{\langle cE \rangle}. \tag{2.3}$$

Here $\langle \cdot \rangle$ denotes time average and $\| \cdot \|$ denotes the norm of the vector. This diffuseness value is in range $[0, 1]$ with value of one implying totally diffuse sound and value of zero the complete opposite.

2.2 Sound propagation

Sound propagation through air is not as simple as with light as the particles are much larger and sensitive to the surrounding conditions. This can produce surprising effects like bending of the wave in some situations. However, in controlled situations[1], it is possible to use some assumptions and work with simpler theories.

[1] A normal room where the air pressure and temperature are relatively constant is already controlled enough.

One method is to model the sound sources with point sources. This means that the source is infinitely small point and radiates waves equally to all directions from it (Fig. 2.1(a)). Due to the symmetry of the wavefront the rising equations tend to be simpler to solve. A second useful simplification is to think that after a long enough distance, the radial wavefront from a point source can be modelled with a simple plane wave (Fig. 2.1(b)). This again effectively reduces complexity.

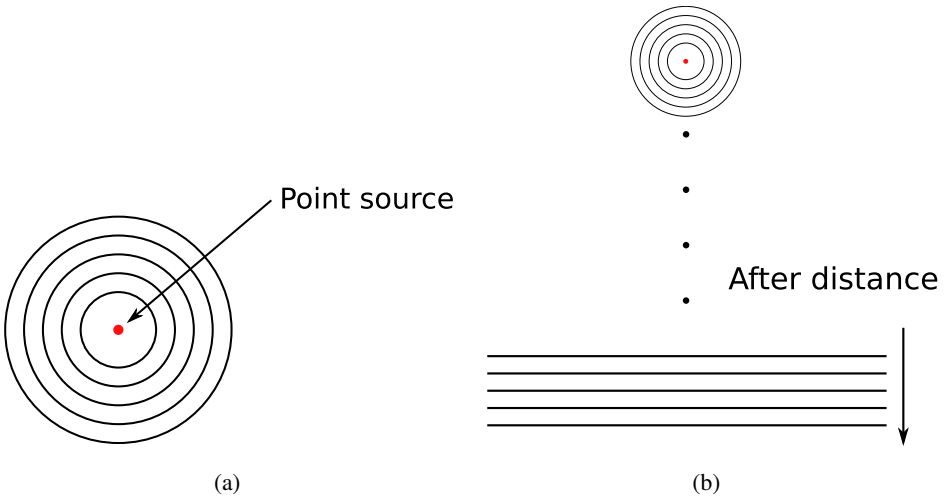


Figure 2.1: (a) Sound radiates evenly from a point source in and (b) after a distance it can be estimated with a plane wave

As the sound waves propagate away from the source, the sound energy is divided constantly to a larger area. In free field[2], the sound pressure is inversely proportional to the distance traveled. This means that if distance is doubled then the sound pressure will drop in half.

2.2.1 Reflections

If sound waves come to contact with a surface, one of three things can happen. The sound wave can be reflected, it can be absorbed by the surface or it can transmit through the surface. In most situations all of them happen with varying degrees. Reflections work similarly to light reflections. If the surface is smooth and flat then the reflection is just like from a mirror (Fig. 2.2(a)). On the other hand, if the surface is rough when compared to the wavelength of the sound wave then the reflection will be diffuse and sound wave will spread to all directions from the surface (Fig. 2.2(b)).

[2]Free field is a condition where there is nothing hindering the propagation of sound.

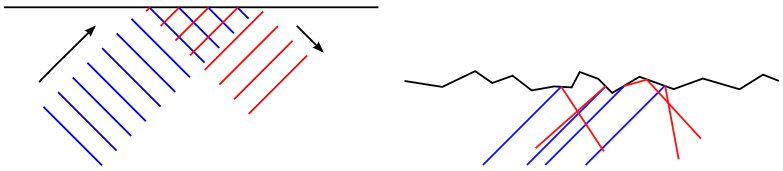


Figure 2.2: (a) Sound reflects evenly from a flat surface. (b) Sound reflects diffusely from a rough surface.

2.2.2 Reverberation

Reverberation is the combined effect of all reflections in the listening space. As the sound moves from the source to all directions, it will encounter walls and other objects which reflect the sound wave. If the sound waves are measured in one point, a listening point, then usually the sound moving on the direct path from the source to the listening point will arrive first. After that, a number of reflected sound start to arrive. First, only a small number of reflections arrive but the number of arriving reflections rise through time rapidly. If an impulse is sent from the sound source, then the measured sound pressure at one listening point in the room can be seen in Fig. 2.3. As it can be seen, the first arriving sound is the direct sound. Then a small number of independent reflections arrive. This part is called early reflections. After a while, the amount of independent reflections is so large that it is impossible to separate them anymore. This stage is then called late reverberation. Actually, this whole thing is called the room impulse response[3] and defines the properties of the room quite well.

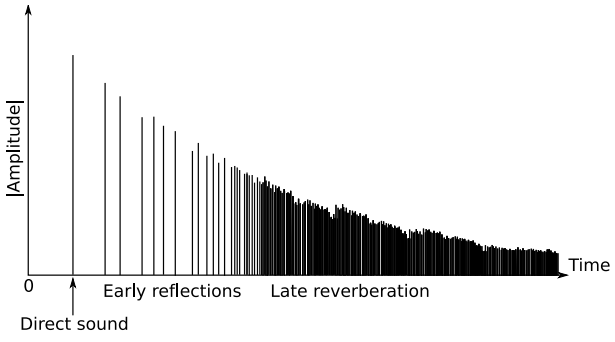


Figure 2.3: Sound pressure at the listening point when the space was excited with an impulse.

[3] More of impulse responses in chapter 4.

Chapter 3

Hearing and Psychoacoustics

This chapter will inspect how the hearing sensation functions. First, there will be a quick description of the anatomy and the physiology of the auditory system. After that, a more in-depth review will be performed on the topic of psychoacoustics which focuses on the sensation of hearing. Psychoacoustics in itself is a widely studied subject and can be quite complex to completely understand. In the context of this thesis, only necessary topics will be described in addition to the basic knowledge. A more comprehensive study can be found in a book by Moore (1995b).

3.1 Auditory system

To understand some of the psychoacoustic properties of the human hearing it is necessary to first study the anatomy and the physiology of the ear. A natural way to do this is to follow the path from the outside of the head all the way to the hair cells which finally convert the vibrations to neural impulses. It is wise to refer to the Fig. 3.1 throughout this trip.

The trip starts from the outside, from a distance of the head. The sound waves are first affected by the head and the shoulders. The next step is the pinna which acts as a complex resonator modifying the spectrum of the sound based on the direction from which the sound arrives. Then, the sound waves enter the ear canal which is a tube leading to the tympanic membrane. This tube also amplifies some of the frequencies due to the natural resonances. This ends the outer-ear-part of the trip.

Next, it is time to move to the middle ear. This is done through the tympanic membrane, or

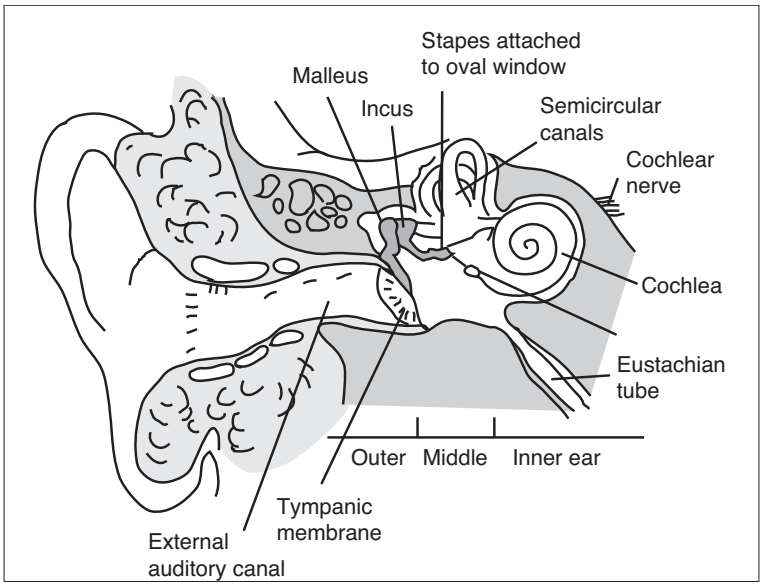


Figure 3.1: A cross section of the human ear. Different parts of the ear can be seen in it. Adapted from Karjalainen (2009).

the ear drum, which is connected to the malleus. The function of the tympanic membrane is to transform the pressure changes in the outer ear to vibrations in the malleus. Malleus is one of the three ossicles, which are three small bones found in the human ear, with incus and stapes being the other two. Stapes is attached to the oval window of the cochlea and incus connects the other two. As a team, ossicles transform the sound-induced-movement of the tympanic membrane to a form which is suitable for entering through the membrane in oval window to the fluid environment of the cochlea. Middle ear also contains the eustachian tube which is important for equalizing the pressure on both sides of the tympanic membrane.

The trip now continues to the inner ear. The sound transformed to vibration by the ossicles now enters the cochlea from the oval window and moves towards the helicotrema. A simplification of the cochlea can be seen in Fig. 3.2. The fluid vibrations then excite the basilar membrane. The excitement spot changes smoothly depending on the frequency with highest audible frequencies exciting, or resonating, more at the beginning of the membrane. The resonant frequency then lowers with the distance traveled through the cochlea and the lowest audible frequencies resonate at the end of the cochlea. This place dependency produces the ability to divide the signal to its frequency components. Finally, the vibrations move back to the entry of the cochlea and out through the round window.

The trip is almost at end as the basilar membrane has already been excited. With the aid

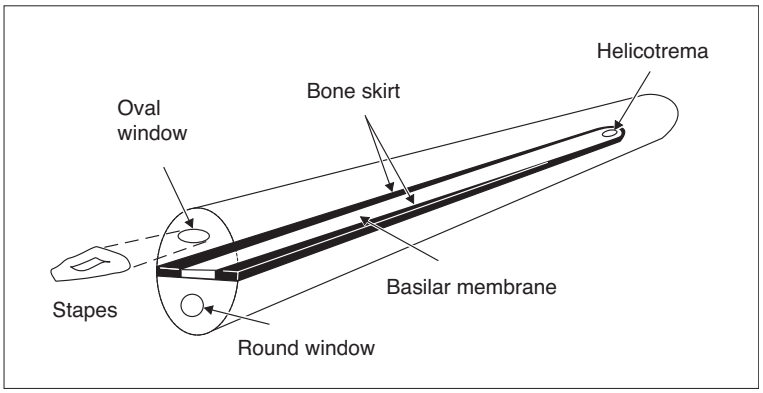


Figure 3.2: A simplification of the cochlea. The stapes connect to the oval window through which the sound travels to the cochlea. Sound then travels to the end of the cochlea exciting the basilar membrane on the way. Adapted from Karjalainen (2009).

of a cross section of cochlea in Fig. 3.3, it can be studied in more detail. As the basilar membrane vibrates, the fluids in the Scala media to also move. Inner hair cells then detect this fluid movement and generate neural impulses which then travel a complex path to the cortex to produce the hearing sensation (Goldstein, 2002). Outer hair cells, on the other hand, mainly receive information from the brain and act as a pre-amplifier to enhance hearing.

This ends the trip but there is still a one more unexplained part in Fig. 3.1. The inner ear contains a part called semicircular canals which are very important for keeping balance.

3.2 Psychoacoustics

As it was previously mentioned, Psychoacoustics is the study of the hearing sensation. This study is usually conducted through subjective listening tests. Psychoacoustics is also an important topic in the context of this thesis as the algorithms presented later on, are derived based on the psychoacoustic properties of the human hearing. Even though all the following phenomena work together to produce the hearing phenomenon, it is prudent to study them separately.

For these following sections, it is necessary to define the concepts of auditory and sound events. Sound events are separate events where sound arrives to the physical part of the

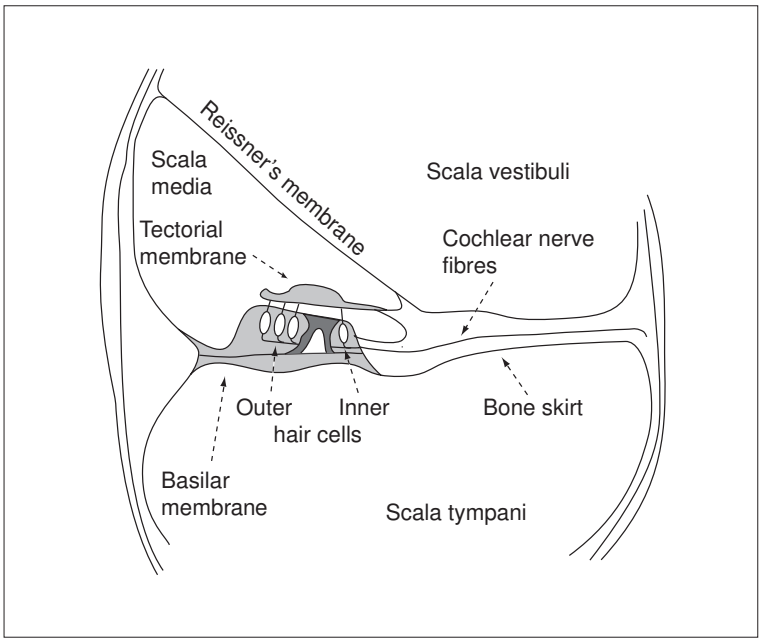


Figure 3.3: A cross section of the cochlea. Inner hair cells detect the movement of the fluid in scala media caused by the vibration of the basilar membrane and produce neural impulses. Adapted from Karjalainen (2009).

auditory system. Auditory events, on the other hand, are events which are perceived in the cortex.

3.2.1 Critical bands

The frequency resolution of the hearing is usually represented with the concept of critical bands. The idea is that inside one critical band only one auditory event can be perceived. If multiple sound events happen inside one critical band then a combination event is heard instead. Also, the critical band is centered on the sound event instead of being at a constant position. This is quite logical when one remembers that the basilar membrane is indeed a membrane and thus cannot vibrate just in an infinitely small point.

There are different schools in how to model the critical bands. A classical critical band is measured as follows. The subject alternates between listening to a narrow band noise and another noise with a variable bandwidth but constant sound pressure level. The subject is then asked to change the variable noise level so that both noise signals sound subjectively equally loud. The result is that when the bandwidth is increased, the loudness first stays

at the same value and after a certain threshold, suddenly starts to increase. This threshold defines the critical band and produces Eq. 3.1 for calculating the bandwidth related to the center frequency (Zwicker et al., 1957).

$$\Delta f = 25 + 75 \left[1 + 1.4 \left(\frac{f_c}{1000} \right)^2 \right]^{0.69} \tag{3.1}$$

This equation was formed based on listening test results. Frequency scale related to this classical method is called Bark scale.

An alternative method of defining critical bands is called Equivalent Rectangular Bandwidth (ERB). This method uses masking noise on both sides of a test signal to remove the possibility that hearing reacts to the test signal with hair cells outside the critical band. By changing the bandwidth between masking noises, it is possible to experimentally measure the bandwidth. The bandwidth related to the center frequency is defined as (Glasberg and Moore, 1990)

$$\Delta f = 24.7 + 0.108 f_c. \tag{3.2}$$

The corresponding frequency scale, called ERB scale, results in more and narrower frequency bands than the classical Bark scale.

3.2.2 Masking phenomena

Hearing has an interesting property of masking sound events which are near in time or frequency. These are consequently referred as time masking and frequency masking. Time masking can be thought as an envelope (see Fig. 3.4) around each sound event. If another sound event arrives and its loudness is under the envelope of the previous sound event, then it will not be perceived. Similarly, frequency masking can be thought as an envelope in frequency domain (see Fig. 3.5). Again, sound events which fall under the masking envelope of another sound event, will not be perceived. However, frequency masking is also sensitive to the content of the masking signal and the shape of the mask changes quite a lot based on the signal.

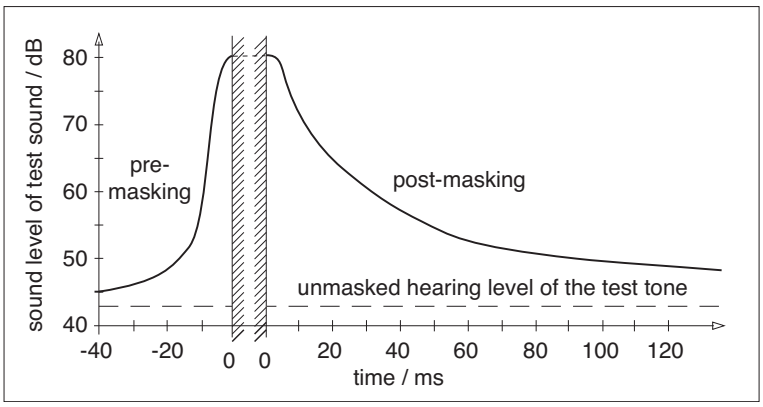


Figure 3.4: The effect of time masking. The shaded area represents the time when the masking sound event is present. In this figure, it is presumed that the masking sound is at least 200 ms long. Adapted from Karjalainen (2009).

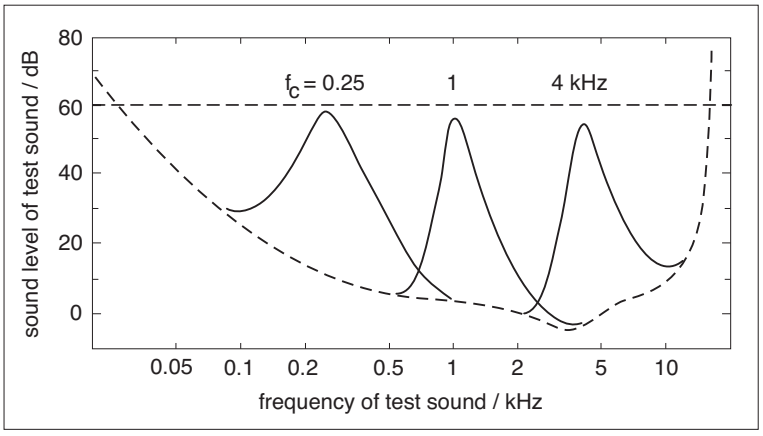


Figure 3.5: The effect of frequency masking with different narrow band noise signals. Notice how the envelope shape changes with the center frequency of the narrow band noise. Different signals have different frequency masks which they produce. With complex music signals, the mask is also complex. Adapted from Karjalainen (2009).

3.2.3 Spatial hearing

The subject of spatial hearing has been studied quite a lot. A comprehensive study can be found for example in Blauert (1997). Another good source is Moore (1995a) in which this section is based on.

Spatial hearing can be divided in two tasks: the localization of sound sources and the perception of spatial impression. Single sound sources are localized with the combination of Inter-aural Time Difference (ITD) cues, Inter-aural Level Difference (ILD) cues and spectral cues. Spatial impression is the impression a listener perceives about the physical properties of the listening (simulated or not) space, like the size, reverberation and envelopment. It is produced as a combination of ILD cues, ITD cues and, in traditional view, Inter-aural Cross-Correlation cues (IACC).

Sound source localization

For this section, it is first necessary to explain ITD and ILD properly. Inter-aural Time Difference is defined as the time difference for a sound signal between its entry to the left and the right ear. It is caused by the effect that if sound is not arriving from the median plane, then there is a different distance from the sound source to one ear of the listener compared to the other. As the speed of sound is limited, this directly translates to a time difference. Notable is the fact that hearing analyzes ITD from the signal differently on different frequency ranges. At low frequencies, ITD is directly the phase difference between the waveforms while at high frequencies the delay between the envelopes of the waveforms is used.

Inter-aural Level Difference is defined as the level difference of a sound signal on entry to the left and the right ear. This is also caused by the listeners head by attenuating the signal moving to the farther ear. Both of these effects can be seen in Fig. 3.6.

Lateral sound source localization is produced primarily by ITD and ILD cues. This process is called the duplex theory (Rayleigh, 1907) and states that on low frequencies, ITD cues are the dominant method of localization and at higher frequencies, ILD cues are dominant. The dividing frequency has been found to be around 1500 Hz (Feddersen et al., 1957). However, it has been found that the strict duplex theory does not hold and ITD cues also affect the perception on the higher frequencies. Still, duplex theory is applicable in certain situations (Hafter, 1984).

The combination of ITD and ILD cues are only able to detect where the sound is coming from in left-right direction. To detect successfully up-down and front-back direction, spectral cues are needed. They are direction dependent radical changes[1] in the sound spectrum that are produced by the direction dependent filtering of the pinna and reflections from the

[1] A radical change here is a dip or spike in the spectrum.

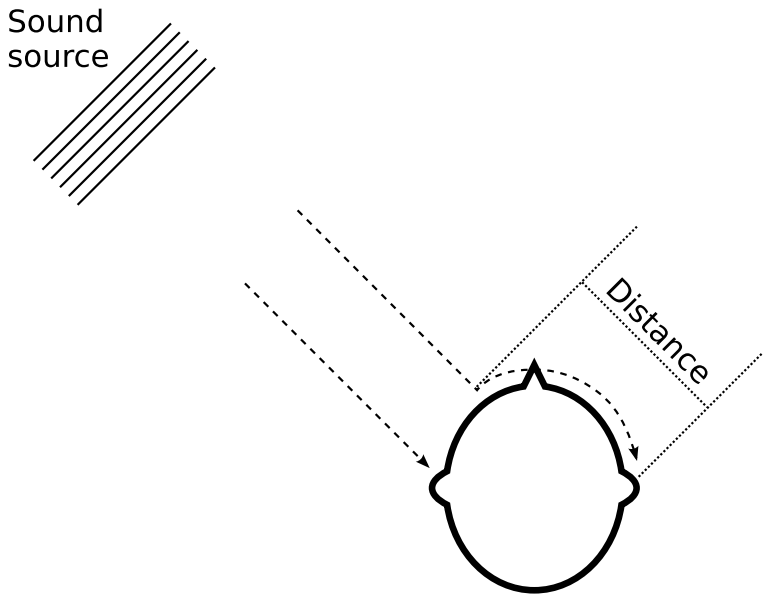


Figure 3.6: The effect of head to an arriving sound. Signal to the farther ear is delayed and attenuated by the listeners head.

shoulders. This, with the addition of instinctive turning of head towards the sound source, enables full three dimensional localization of sound source direction.

Additionally, the hearing has some restrictions in terms of localizing multiple concurrent sound events. The aforementioned frequency resolution in terms of critical bands is one and affects the perception of multiple sound events if they fall on the same critical band. Even more important is the property of summing localization. If multiple sound events arrive close to each other in time (in the range of 0 to 1 ms) then the localization is based on the superposition of those signals based on the inter-aural differences (Blauert, 1997).

However, the accuracy of localization is limited and listening tests have shown that the accuracy changes based on the actual source direction. Without further in-depth study, figures for horizontal (Fig. 3.7) and vertical (Fig. 3.8) localization accuracy are presented (Blauert, 1997).

Perception of auditory distance, that is the perceived distance to the sound source, has been studied less than the direction localization. Grantham (Moore, 1995a) notes that four cues have been identified for distance perception and they are as follows.

1. Sound pressure level – greater means closer

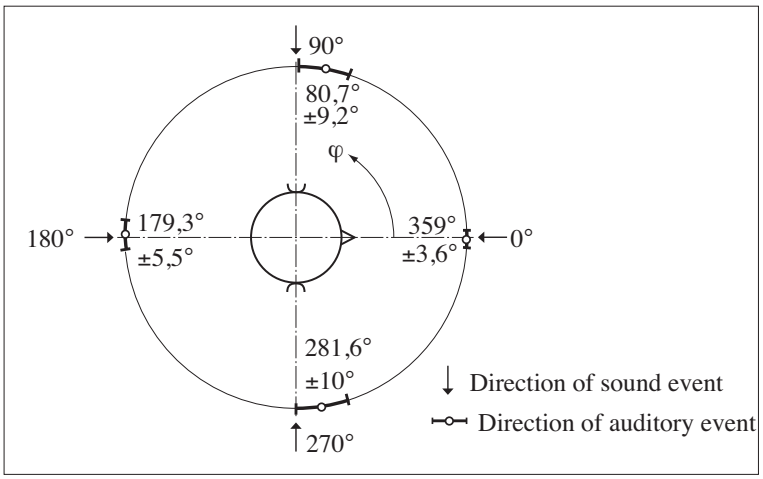


Figure 3.7: Accuracy of localization on horizontal plane.

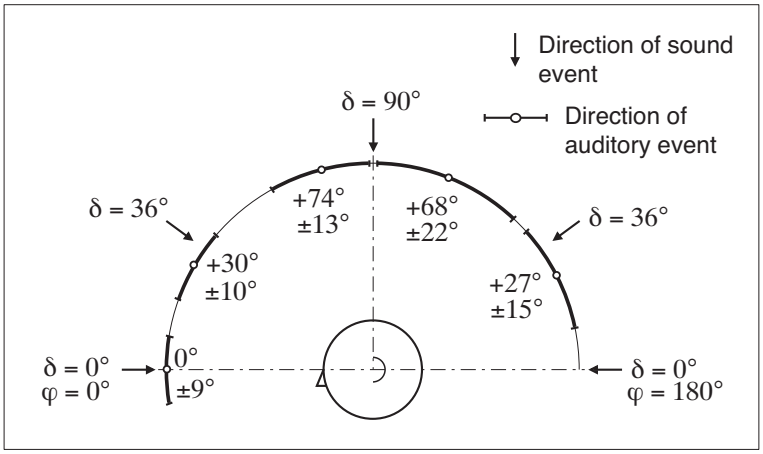


Figure 3.8: Accuracy of localization on vertical plane. Notice that the events are opposite compared to the horizontal accuracy figure.

2. Direct to reverberant energy ratio – greater means closer

3. Spectral shape of the signal at longer distances (over one meter) – more high frequencies means closer

4. Binaural cues at close distances (under one meter) and off the median plane – greater ITD or ILD means closer. However, the evidence for this is inconclusive (Blauert, 1997).

Spatial impression

Spatial impression is produced by the localization cues but, in traditional view, is also affected by Inter-aural Cross-Correlation (Blauert, 1997). IACC is the cross-correlation between the signals entering the left and the right ear. In addition to the time difference, it also contains information about spectral differences between signals arriving to the ears. High values of correlation mean that the signal is from a single, accurately localizable source. On the other hand, low values mean that the sound source cannot be localized accurately and the sound seems to enveloping the listener. How this transforms to a spatial impression is that lower IACC values mean that there are multiple versions of the original source signal present. This is exactly what reflections and reverberation is and IACC is thus connected to the spatial properties of the listening space.

However, the effect and existence of IACC has been questioned in auditory processing. Instead, a proposal has been made, that the spatial impression is generated from the fluctuation of ITD and ILD cues (Griesinger, 1997). This proposal is important in the context of this thesis as Directional Audio Coding does not assume anything about IACC and still aims to reproduce spatial impression.

Precedence effect

Another important topic of spatial hearing is the localization of sound source when there are reflections and reverberation present. With aforementioned localization cues, it would be quite hard to directly localize the sound source accurately as there are usually multitude of reflections present which could simply be new sources. However, hearing has a property of favoring the direction of the first sound from a multitude of similar signals. This property is called the precedence effect (Yost and Gourevitch, 1987). Precedence effect has been studied and it has been found that there is a certain time delay after which the first sound dominates the direction perception. Blauert (1997) presented that, with delays between 1 ms and 30 ms, precedence effect affects the direction perception. If the delay is smaller, then the localization is based on the combination of the signals. If the delay is larger, then a separate echo will be heard. This latter echo threshold is variable and depends on the signal. Impulsive signals have a smaller echo threshold whereas with continuous signals the threshold is larger.

Visual cues

Spatial hearing is not defined only by auditory cues. Visual cues have a significant impact on the perceived direction. If the listener can see a clear sound source then there is a tendency that the sound is perceived to come from it if auditory cues do not clearly tell otherwise. This is called the ventriloquism phenomenon. More of this can be found for example in Radeau (1994).

3.2.4 Pitch, loudness and timbre

Pitch, loudness and timbre are important terminology used when discussing hearing. Pitch is the subjective quality of how "high" or "low" the perceived sound is. It is often related to the fundamental frequency of the harmonic sound event but is not always the same. Similarly, loudness is the subjective quality of how "loud" or "soft" the perceived sound is. Again, this value is related to the physical quantity of sound pressure level but is dependent on signal content. The final term, timbre, is a more complex one but is still simply defined as the thing which separates two sound events with the same pitch and loudness from each other. It can be related to the time-dependent spectrum but again does not fully correlate with it.

Chapter 4

Signal Processing

This chapter presents elementary signal processing theory. This theory is the framework on which Directional Audio Coding, like all signal processing algorithms, is built. Explanations will be based on graphical examples when possible and there will not be any derivation of formulas here. More comprehensive knowledge of signal processing can be found in Sanjit K. Mitra's book Digital Signal Processing (Mitra, 2006) in which this chapter is largely based on. Another good source for frequency domain processing is the freely available book by Smith (2008a). Also, the explanations are given with audio signals in mind where possible.

4.1 Sound signals

As it was presented in chapter 2, the sound is essentially pressure changes moving through air. If the pressure or any other property is measured in one point depending on the time, then the measurement result is called a signal. For sound, this measurement is performed with a microphone which transforms sound pressure to an electric signal. Alternating pressure is transformed to alternating voltage which is then easy to transmit or modify. Loudspeaker is the inverse pair of a microphone and transforms alternating voltage to alternating pressure.

Signals can be inspected with a multitude of representations. The most definitive representation is the mathematical representation where the signal is given as a formula. For example, the formula for a sine wave is shown in Eq. 4.1 and the formula for an ideal

square wave is shown in Eq. 4.2.

$$x_{sine}(t) = A\sin(\omega t) \tag{4.1}$$

$$x_{square}(t) = A\frac{4}{\pi}\sum_{k=1}^{\infty}\frac{\sin((2k-1)\omega t)}{2k-1} \tag{4.2}$$

These formulas are simple but some of the mathematical formulas are hardly illustrative. Instead, it is much more useful to show a part of the signal as a figure. In Fig. 4.1 a few periods of both aforementioned signals can be seen. This shows a clear representation of the signal which is understandable with a quick look.

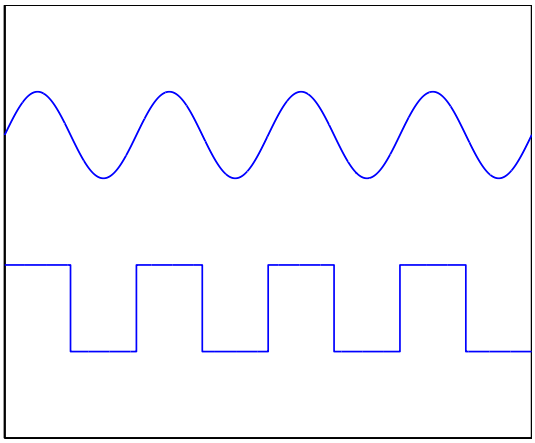


Figure 4.1: Four periods of a sine wave (the upper one) and an ideal square wave (the lower one). Both have same amplitudes.

A third option of representing signal is to measure a series of time-value pairs which define a discrete sequence of signal values. It is not as accurate as the mathematical representation but proves to be useful like it is shown in the next section. Usually time values are taken (sampled) with a constant step size in time so that there is no need for presenting time value for each pair. Instead the signal is represented as series of values and one value is anchored to a certain time point to define the series fully.

4.1.1 Digital signals

Previously mentioned electric signals are usually dubbed with a term analog signals when they represent changes in some other quantity like the sound pressure in case of sound signals. This term is created from the process how the microphone converts continuous pressure changes to continuous voltage changes in the electric circuit. Thus, the electric voltage is analogical to the sound pressure.

The counterpart for analog signals are digital signals. The difference is that when analog signals have infinite time and value resolution, digital signals have a discrete number of time moments where it has a value. Also the number of possible values is limited to discrete steps. One could think that there is no advantage in reducing information by changing it to digital form but there are a lot of advantages. The most important advantage is that the information reduction also reduces space needed to store and transmit the signal. This enables studying and modifying the signal with computers. Another advantage is realized when there is a need to transmit the signal as all transmission lines are susceptible to noise. With a discrete amount of known possible values for the signal, it is possible to create an error correction algorithm which can reconstruct the signal even after it is corrupted by some amount of noise. This effectively equals noise-free signals if the signal can be reconstructed.

AD- and DA-conversion

Analog signals can be converted to digital signals by sampling and quantizing the signal. This is called analog to digital (AD-) conversion. This process is illustrated in Fig. 4.2. The inverse process is called digital to analog (DA-) conversion and applies interpolation[1] to reconstruct the continuous analog signal from the digital samples. Sampling means that a sample of signal is taken regularly to represent the signal during the time interval called sampling interval. Quantizing then means that there is a discrete amount of possible values which the sample can have. Quantizer rounds the sample's value to the nearest possible value. This rounding, however, produces some distortion to the signal. In DA-conversion, interpolation tries to invert the process as well as possible by calculating values in between samples from the samples. There are multiple methods to interpolate and they give different results. All in all, it is clear that one of the issues defining digital signal processing system quality is the quality of the AD- and DA-converters.

[1] Interpolation is a method of calculating values in between known values using the known values.

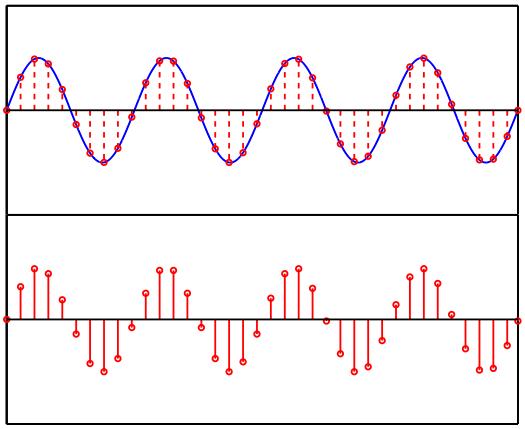


Figure 4.2: Sampling and quantization of a sine wave. The upper waveform shows the original sinusoid and the time samples taken from it. The lower waveform is the digital signal received after the time samples are linearly quantized.

Limitations

Digital signals, however, have some restrictions. As it was previously mentioned, in AD-conversion, values are quantized to discrete possible values. This produces noisy distortion depending on how much the quantized value differs from the original value. Usually, the number of possible values is selected high enough or matched to the signal with various methods to reduce the level of quantization distortion under audible level.

Sampling interval also produces a restriction as there is a finite time resolution available. Simply put, information between two consecutive samples is essentially lost and the smallest period possible to represent correctly is the one spanning exactly three consecutive samples. As the frequency is the inverse of the period, this means that the frequency is bound from above. This limit frequency is exactly half of the sampling frequency f_s and often called Nyquist frequency. Frequencies higher than this border do not convert correctly to digital domain. More of this will be discussed in section 4.2.1.

4.2 Frequency domain

In previous sections, signals were studied as values depending on time. This representation is also called the time domain representation.

In some cases, it is easy to identify the signal from a time domain representation (i.e. sine wave) and estimate its parameters. However, with for example a speech signal (Fig. 4.3), the signal is too complex to be identified by simply looking at the waveform.

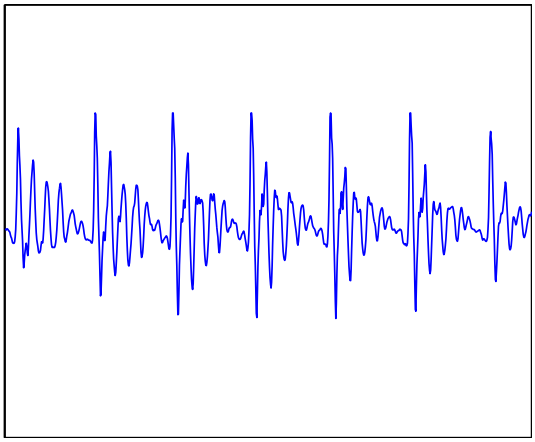


Figure 4.3: A sample of vowel /a/ from the word "kaksi" in finnish. Periodicity of the signal can be seen but it would be hard to tell from this form what this signal actually represents.

In many cases more information can be seen from the frequency domain representation. In frequency domain representation the signal is presented in a form where the properties of the signal are given depending on frequency instead of time. This representation can be formed from the time domain representation with the aid of Fourier transform. Fourier transform for continuous (analog) signals is defined in Eq. 4.3 and corresponding transform for discrete (digital) signals is defined in Eq. 4.4.

$$X(\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{4.3}$$

$$X(k) = \mathcal{F}_d\{x(n)\} = \sum_{n=0}^{N-1} x(n)e^{-jk2\pi\frac{n}{N}}dt \tag{4.4}$$

The result is referred as the spectrum of the signal. Note that the frequency domain signal is denoted here with a capitalized letter. This is a convention used widely and will also be used in this thesis. Also note that the discrete Fourier transform is calculated from a limited amount of values called a "block" with a length of N samples. The assumption is that this block repeats in time endlessly.

The assumption behind Fourier transform is that any signal can be represented as a linear combination of sinusoidal signals with different frequencies. In spectrum, the amplitude and phase parameters of the sinusoids are shown depending on the frequency. In audio processing context, the amplitude is often more interesting as human hearing is more sensitive to amplitude than phase.

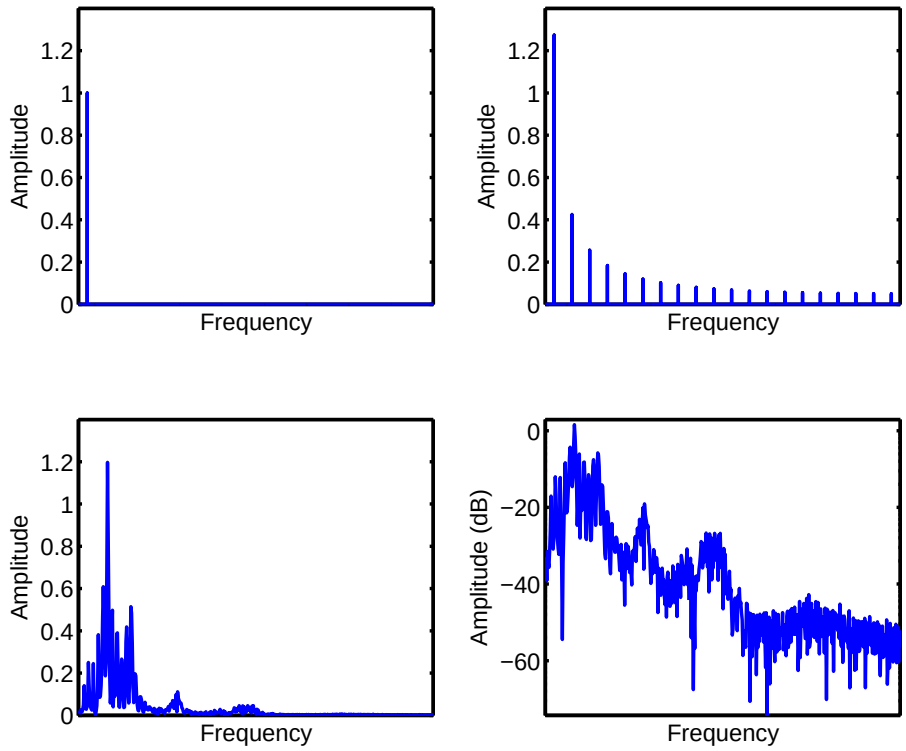


Figure 4.4: Amplitude responses of a sinusoid (up-left), a square wave with a same fundamental frequency as the sinusoid (up-right) and the previous (Fig. 4.3) speech signal (down-left). Lower right figure contains the same speech signal but amplitude axis is logarithmic and thus the amplitude envelope is more clear to see.

In Fig. 4.4 the amplitude spectra of the sinusoid, square wave (Fig. 4.1) and speech (Fig. 4.3) signals are presented. The amplitude spectrum is achieved by taking the absolute value from the complex spectrum. As it can be seen, the sinusoid signal has only one non-zero value in the figure as there is only one sinusoid from which the signal is constructed. Square wave, on the other hand, is constructed from multiple harmonic sinusoids with a frequency

dependent amplitude and thus there are multiple non-zero values in the amplitude spectrum. The spectrum of the speech signal, however, is more complex. In this case, there is a large amount of sinusoids constructing the signal and individual parameters offers little information. Instead, the amplitude envelope is more interesting. It happens to be so that human hearing recognizes different vowels of speech based on the shape of the amplitude envelope. This envelope is even better seen when the amplitude scale is made logarithmic which is customary with certain signal types.

Inverse transforms, which return the signal from frequency domain to time domain, are similarly defined with formulas 4.5 and 4.6.

$$x(t) = \mathcal{F}^{-1}\{x(t)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}dt \tag{4.5}$$

$$x(n) = \mathcal{F}_d^{-1}\{x(n)\} = \frac{1}{N}\sum_{n=0}^{N-1} X(k)e^{jk2\pi\frac{n}{N}}dt \tag{4.6}$$

Discrete versions of the Fourier transform are usually called discrete Fourier transform (DFT) and inverse discrete Fourier Transform (IDFT). These transforms are essential in digital signal processing and there exists an especially fast algorithm for calculating them called fast Fourier transform (FFT).

4.2.1 Aliasing

As it was previously mentioned in section 4.1.1, with digital signals, it is impossible to represent frequencies higher than the Nyquist frequency $\frac{f_s}{2}$. The problem is that some operations (especially the non-linear ones) produce new frequency components of which some would on higher frequencies than the Nyquist frequency if it was possible. The result is that frequencies larger than the Nyquist frequency will mirror around that border frequency producing an alias. This results in usually unwanted frequency components which are often quite audible. The pre-emptive solution is to ensure that no aliased components are created. However, this is not always possible and aliasing suppression methods have to be applied. These methods, however, are out of the context of this thesis.

4.2.2 Windowing

As it was mentioned in section 4.2, DFT is applied to a block of signal which it presumes to repeat periodically in time. The problem is that, the first and the last samples in the block do not necessarily have values close to each other. This difference or more precisely, discontinuity, can result in radical (and often incorrect) differences in frequency domain representation. To alleviate this problem a window function may be applied to the signal block. This window usually forces both ends of the block near zero and attenuates parts which are far from the center of the block. This removes discontinuities but at the same time, values in the frequency domain become less accurate as some of the frequency resolution is lost.

Different window functions have been proposed for use with the DFT. One of the most used window functions is the raised cosine window. Two popular forms of it exist, the Hann window formed with equation

$$w(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right), \tag{4.7}$$

and the Hamming window formed with equation

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right). \tag{4.8}$$

They have slightly different properties but essentially perform similarly. They also have useful properties when used in real-time signal processing (further explained in section 4.4.3).

4.3 Digital systems

Digital systems, like the name says, are systems which process digital signals. They can be thought as a black box[2] which takes in an input signal $x(n)$, processes it and outputs a processed signal $y(n)$. Essentially, it is only needed to know how the system modifies the signal, not what it actually contains.

[2]This theory applies similarly to continuous time analog signals but it will not be discussed here.

It happens that there is a sequence of values which defines exactly what the system does. This is called the impulse response of the system and is usually denoted with $h(n)$. It is produced in output of the system when input sequence contains a single value of one at one time moment and after that only zeroes. The Fourier transform of the impulse response is also a special case and called the frequency response of the system and often denoted with $H(k)$. It shows how each frequency component of the input signal is affected when it passes through the system.

4.3.1 Convolution

The output signal of a system can be calculated from the input signal and the impulse response of the system with an operation called convolution. If $x(n)$ is the input signal, $y(n)$ is the output signal and $h(n)$ is the impulse response of the system then

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{\infty} x(i)h(n-i) \tag{4.9}$$

is true. Note that formally it is stated that this operation is calculated with all values of the signals, thus the infinite boundaries for the sum. However, usually at least one of the terms is of finite length and thus the boundaries reduce to the values where the product is non-zero.

The formula is quite simple but it might be hard to visualize what is really happening. In Fig. 4.5 it can be seen that in convolution one of the signals is inverted in time and then slid over the other one. Result is then achieved by multiplying together values which are at the same time spot and summing the multiplications together. This result is then stored to the spot where the rightmost value of the sliding signal is currently. With finite length digital signals the length of the result is exactly the sum of the lengths of the input signal and the filter impulse response minus one.

Convolution in itself is quite complex operation as it requires many multiplications and additions. However, there happens to be a certain advantageous property between convolution and Fourier transform. That is called the convolution theorem and is formulated as

$$\mathcal{F}\{x(n) * h(n)\} = \mathcal{F}\{x(n)\}\mathcal{F}\{h(n)\}. \tag{4.10}$$

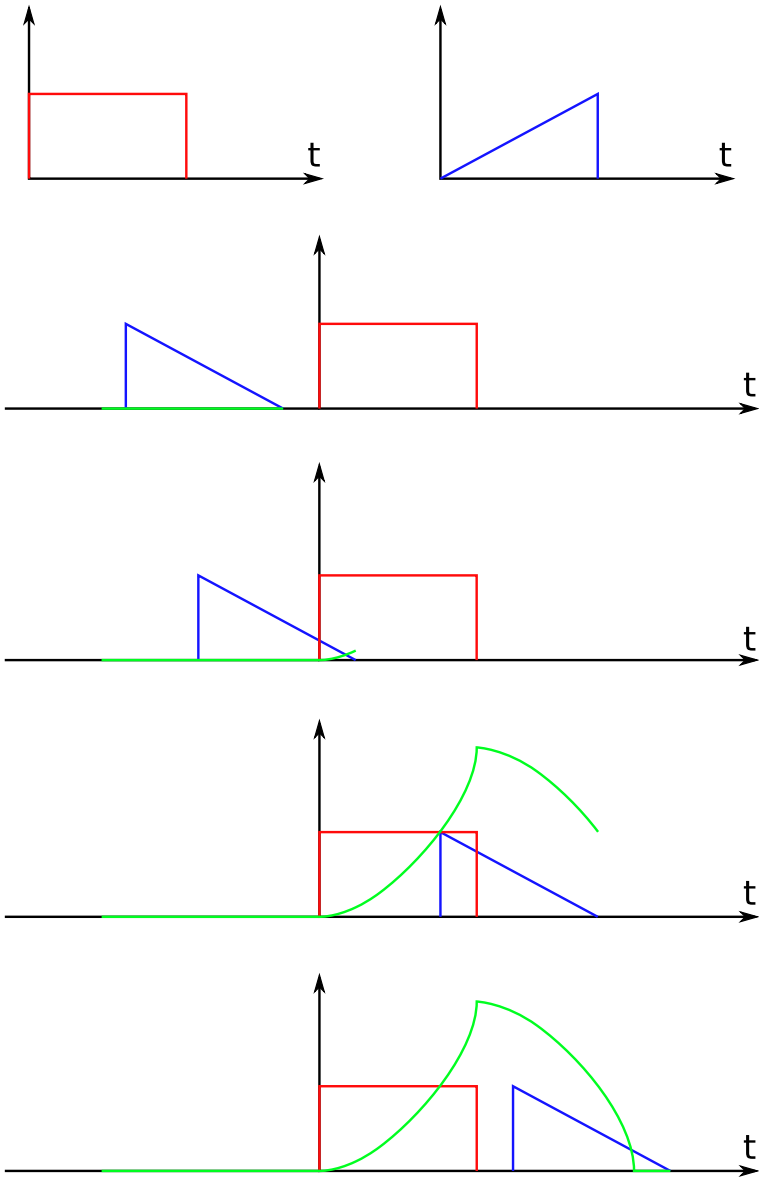


Figure 4.5: The calculation of convolution. Two signals (red and blue) are convolved with each other to produce the result signal (green). First one of the signals is inverted in time and then slid over the other signal. At each time moment, the values at same time moment are multiplied and then a sum of the multiplication results is calculated. The result is stored to the position where the first value of the sliding signal is moving.

This means that convolution turns into a multiplication operation in frequency domain. This is also true vice versa and convolution in frequency domain is equal to multiplication

in time domain. There is an important note which has to be made here. This equation works exactly like this only with analog signals. With digital signals the corresponding operation is actually circular convolution which is caused by the assumption of DFT that the time domain signal is periodic from the block boundaries. To calculate linear convolution with circular convolution, it is necessary that zeroes are added to the end of the input signal and the filter impulse response before transforms so that their lengths are at least the same as the previously mentioned length of the convolution result.

4.3.2 Linearity and time-invariance

An important property often desired from a digital system is linearity and time-invariance. They are formally defined with Eq. 4.11 for linearity and Eq. 4.12 for time-invariance.

$$h\{ax_1(t) + bx_2(t)\} = ah\{x_1(t)\} + bh\{x_2(t)\} \tag{4.11}$$

$$y_1(t + t_0) = x_1(t + t_0) \quad \text{, when} \quad y_1(t) = x_1(t) \tag{4.12}$$

What these formulas actually mean is more easy to understand than it might seem. In linear system the impulse response does not change depending on the input signal. In time-invariant system the impulse response does not change depending on time. When a system is both linear and time-invariant (LTI) then the system's modifications to input signal are fully known and deterministic which is often desired.

4.3.3 Filters

Digital systems which are specifically designed to change frequency domain properties of the signal passed through them are called filters. They can be thought to work analogically to a real-life water or air filters. Air filters remove larger molecules from air but let the air pass through without effort. Similarly signal filters remove frequency components (like low frequencies) from the signal. As it was specified earlier in section 4.3, the impulse response specifies a system. Fourier transform of the system was called frequency response and is denoted as $H(k)$. Formally, it is defined as a relation of input and output (Eq. 4.13).

$$H(k) = \frac{Y(k)}{X(k)} \tag{4.13}$$

When a transform called z-transform is applied to impulse response instead of Fourier transform then a form similar to the frequency response is received. This form is called the transfer function and can be seen in Eq. 4.14. This form also defines system perfectly and is really useful in digital signal processing. Without further proving[3], it is noted here that frequency response and transfer function are equal when $z = e^{j\omega}$.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=-\infty}^{\infty} b_k z^{-k}}{1 - \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} a_k z^{-k}} \tag{4.14}$$

Here b_k and a_k are the coefficients of the filter.

Another form of defining a digital system is the difference equation which is in time domain. It specifies directly how the next output sample depends on the input samples and other output samples. The form of the difference equation can be seen in Eq. 4.15.

$$y(n) = \sum_{k=-\infty}^{\infty} b_k x(n-k) + \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} a_k y(n-k) \tag{4.15}$$

There is a clear relation between the difference equation and the transfer function and they can be calculated easily from each other.

Finite and infinite impulse responses

With linear digital systems, there are basically two kinds of filters. The first one is called finite impulse response (FIR) filter. Like the name says, the length of their defining impulse response is finite. This means that with a finite length input signal the output signal is finite length and can be calculated in finite amount of time. Additionally, FIR filters are stable, that is their output does not go to infinity if the input does not contain an infinite value. Only

[3] Proofs can be found in for example Mitra (2006)

previously mentioned coefficients b_k are non-zero in FIR filters and they contain exactly the same values as the impulse response $h(k)$ of the filter.

The second kind is called infinite impulse response (IIR) filter. Again, like the name says, the length of the impulse response is now infinite. This property is based on the fact that IIR filters apply recursive filtering. That is, their a_k coefficients contain non-zero values which means that other output values affect the current output. The result is that IIR filters often require less coefficients to produce same effects than FIR filters. However, the flip-side is that IIR filters can be unstable as there is an infinite amount of values added together.

Causality

One property of digital filters is their causality. Causality means that there cannot be any output from the system before there is an input to the system. Essentially this changes the previous general difference equation (Eq. 4.15) to a causal form

$$y(n) = \sum_{k=0}^{\infty} b_k x(n-k) + \sum_{k=1}^{\infty} a_k y(n-k). \tag{4.16}$$

Causality is a needed property when processing has to be done in real-time as there is no knowledge of future values. The result is that usually this causes also a delay to the system which can be significant with high order FIR filters. However, if the signal is fully known beforehand then it is often prudent to apply non-causal filtering which does not incur delay — also known as zero-phase filtering.

Combined filters

If several filters are applied to the signal one after another or in parallel, the composite system has a certain behaviour. If all filters are linear then all normal implications of linearity apply to them. That is, the sequence in which they are applied can be changed and gain can be applied at any stage. Also the transfer functions (or difference equations) of two parallel filters can be added together to receive composite filter with the same end to end response. Cascaded filters combine a bit differently. Their composite filter is achieved when the transfer functions are multiplied together. This means, of course, that a convolution can be calculated between the impulse responses to receive the impulse response of the composite filter. Fig. 4.6 explains this even more clearly.

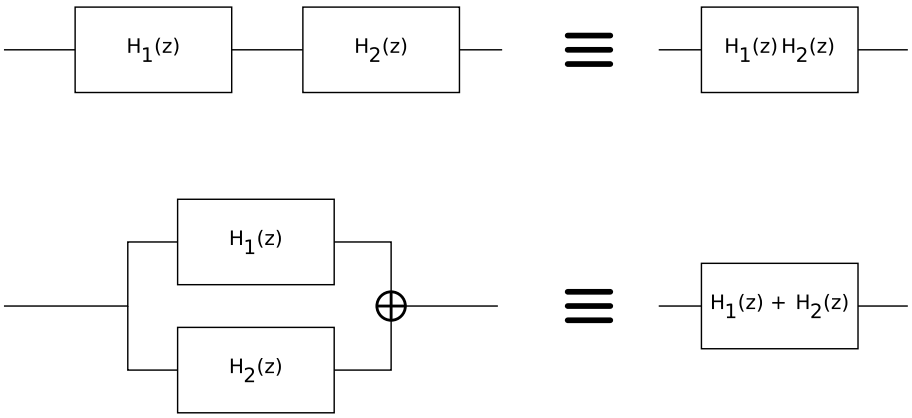


Figure 4.6: Combination of separate filters. Transfer functions of cascaded filters (upper figure) are multiplied together and parallel filters (lower figure) are added together

4.4 Real-time digital signal processing

Real-time digital signal processing produces some requirements for the digital systems used in the processing. First of all, the term real-time implies that the signal will constantly pass through the system. However, it does not require that there is no delay and no digital system is delay free. The allowed amount of delay is decided on a case to case basis. For example, the audio effects used in live performances produce only low delay for performers to be able to play in rhythm. The second requirement is that the used algorithm should be fast enough so that it can be calculated in the amount of time available.

Simple filtering can be done in real-time quite easily with just implementing the difference equations of the filters. More complex systems, on the other hand, usually take advantage of the frequency domain where filtering is generally less tasking to do. However, the use of frequency responses restrict IIR filters out of the picture as their frequency response would also contain an infinite amount of values.

Fourier transform in itself is not well suited for processing long (or infinite) signals as the transform takes the whole signal and transforms that. However, there exist methods which utilize both time and frequency domains and they are called time-frequency methods. One widely used and easily implementable method is called short-time Fourier transform and is described in the next section.

4.4.1 Short-time Fourier transform

Short-time Fourier transform functions just like the name says — it takes a short block of time which is then Fourier transformed to receive the frequency domain representation of that block. In this way even a long input signal can be modified quite easily by multiplication in frequency domain.

However, there are few things which have to be taken in account so that this "block filtering" works well. Firstly, the input signal block and the filter impulse response should be zero-padded correctly like mentioned in the section 4.3.1. Secondly, the choice of correct time window size is vital. This is because there is a trade-off between time and frequency resolution. Better time resolution transforms to worse frequency resolution and vice versa. This limitation is called the uncertainty principle and is formally defined, for example, in a book by Pinsky (2002). In the context of this thesis it is only necessary to understand that the trade-off exists. Usually the window size is matched to the signal which is processed. Finally, to produce correct results the time domain block results have to be combined together correctly. This is further studied in the next section.

4.4.2 Overlap-add

There are two major ways of performing block convolution correctly: overlap-add and overlap-save. Both give the same results and actually only differ a little. In this thesis, the overlap-add method will be used and that is why only that method is described here.

Overlap-add divides signal to smaller non-overlapping blocks and processes these blocks separately. After separate processing, the length of each block is larger than before. To receive the correct result, the current block is added to the position where it resided compared to the beginning of the previous block when it came in as input signal. This whole process is easier to understand from Fig. 4.7 where you can see the whole process done with the aid of frequency domain multiplication.

4.4.3 Windowing STFT

Windowing enhances time accuracy of the processing (and of course takes away some frequency accuracy). It is used for the previously mentioned reasons (see section 4.2.2). However, windowing also causes some restrictions related to the window type and amount of

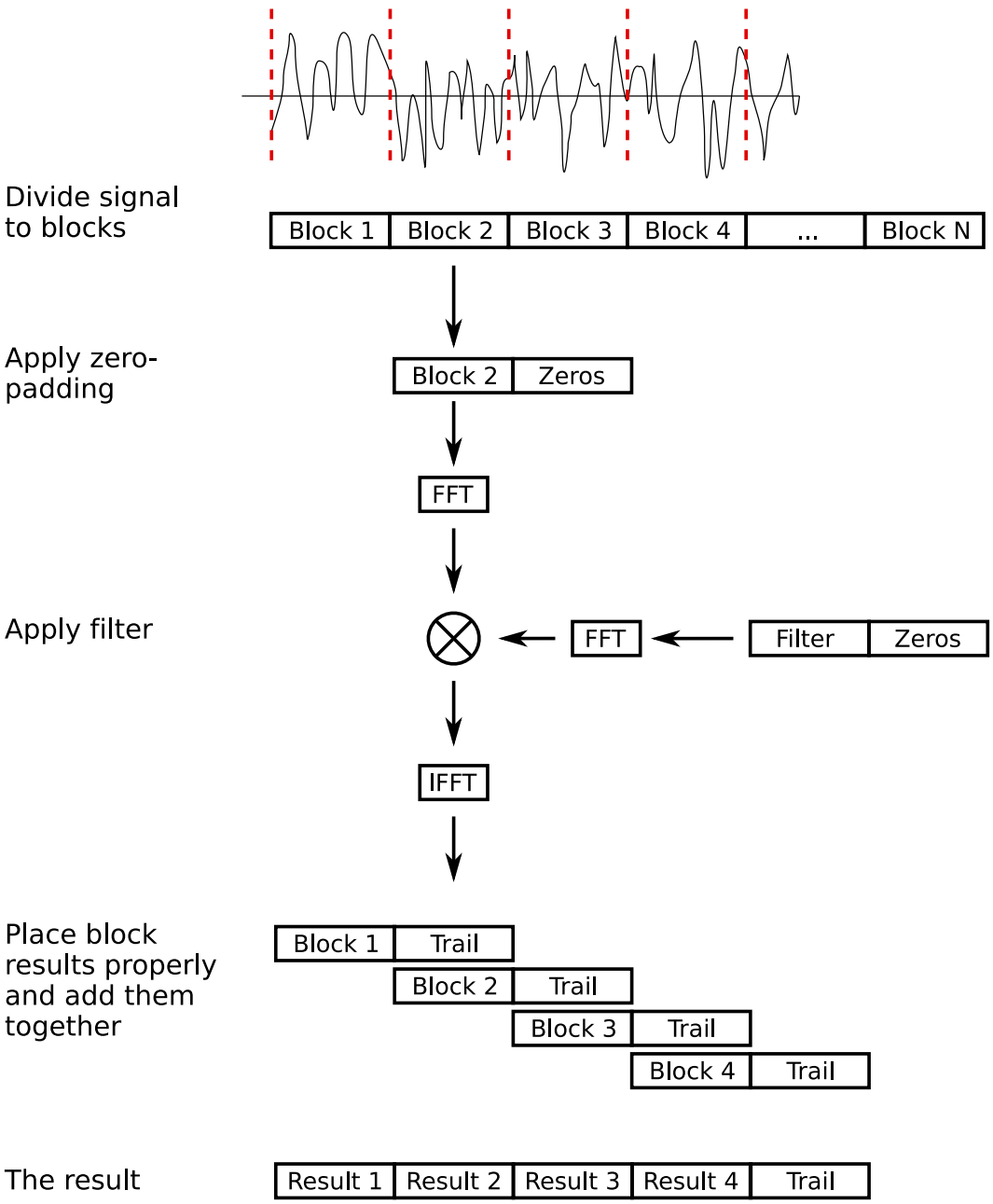


Figure 4.7: The overlap-add algorithm. Done with the added frequency domain multiplication as it is usually done this way to increase efficiency. Note that also the filter is zero-padded to the correct length. Also, the separate block results are always added to the trail of the previous block.

overlap so that overlap-add produces a constant value from the window functions of each block added together. For example with a Hann window, which is a raised cosine window, the correct amount of overlap to receive constant one at the output is 50%. 25% and other division by two cause only constant amplification but problem arises when the overlap is not one of these values. Then the result will vibrate through time and causes an audible tremolo-like effect. More information about different window functions and their correct overlap values can be found in Smith (2008b).

Chapter 5

Sound Reproduction

This chapter describes the reproduction of sound. The focus will be especially on the spatial sound reproduction as the topic of this thesis, Directional Audio Coding, is also a method for that purpose. This chapter is divided to three sections. The first section explains the general idea in sound reproduction. The second section will describe classical systems which have been used already a long time but do not transmit spatial information that well. Finally, the third section will describe modern systems.

5.1 General idea

The general idea of sound reproduction is, as the name says, to reproduce a previously recorded sound (see Fig. 5.1). The aim and the method of reproduction might vary as there are different needs for a telephone conversation and for a concert recording. As this thesis is focused more on the high quality reproduction, it is prudent to approach sound reproduction from that perspective. The aim of high quality reproduction is to reproduce the recorded sound so that the timbre of the original sound is reproduced without distortion and the spatial information (sound source localization, reverberation and spatial impression) is preserved.

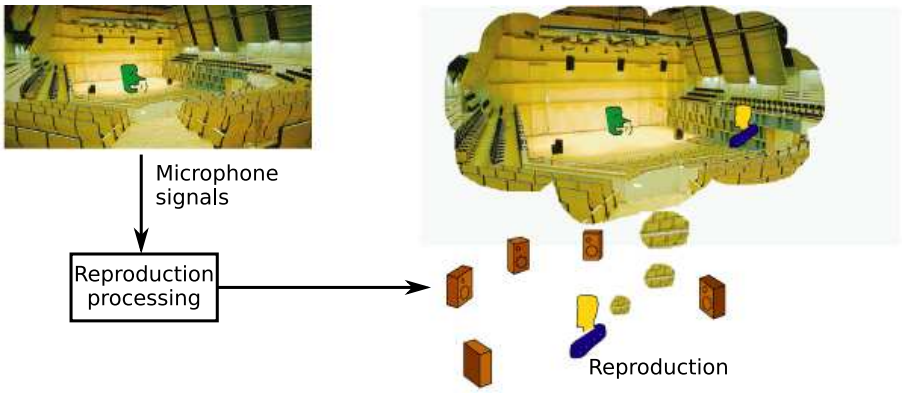


Figure 5.1: The general idea of sound reproduction. First, sound is recorded in a real situation. Then, it is processed with various means and finally sent to various amounts of loudspeakers around the listener. The result is a reproduction of the sound in recorded situation and differs in quality based on the used reproduction method.

5.2 Classical systems

The first sound reproduction system was the phonograph created by Thomas Alva Edison in 1887. It recorded sound on a cylinder which then could be played back. The quality was not that good in modern standards but it was ground breaking invention nonetheless. In terms of sound reproduction, it was a monophonic system producing one sound source to the listener. That means that there is a microphone recording the sound and one loudspeaker reproducing the recorded sound. With a monophonic system, the timbre of the recorded sound can be reproduced quite well. However, the spatial information is lost with the use of only a single channel. Thus, this is not high quality reproduction but is well suited for some applications like telephone.

Later on, a better method of sound reproduction was invented. It is called stereophonic sound reproduction. The formal idea is to use stereographic projection for encoding the relative positions of the sound events recorded. This is done by placing two microphones in a certain positions relative to each other[1] for recording two audio channels. Resulting two-channel signal can be reproduced with number of loudspeakers but the two-loudspeaker version is the most popular one and usually referred as stereo. First stereo transmissions were made already at the late 19^{th} century and it is still the dominant choice of sound reproduction. In terms of quality, stereo also preserves the timbre of the recorded sound

[1]There are different position setups which have their own advantages and disadvantages.

quite well but additionally preserves some of the spatial information and thus is generally more pleasant to listen to.

During technology development of loudspeakers, a lightweight portable version of them was created which is called headphones. Currently, different kinds of headphones are in popular use with portable players. Even though headphones can produce quality sound, they have a problem when compared to conventional loudspeaker systems. The stereo sound listened with headphones does not have the same spatial quality as with loudspeakers. Instead, the sound is perceived to come from inside of the head between the ears which can be annoying.

5.3 Modern systems

As monophonic systems already reproduced the timbre of the recorded sound quite well, modern systems tend to focus on providing a better reproduction of the spatial information of the recorded sound. This is achieved with quite varying methods.

5.3.1 Binaural recording

In binaural recording, the sound is recorded with two microphones placed to the ears of an artificial or a real head. This recording can then be reproduced with headphones so that the recorded left and right ear signals go to the corresponding speakers of the headphones. The advantage is that the head used for recording modifies the sound signals similarly as the listeners head would modify in the real situation. This means that the spatial information is "coded" to the two recorded sound signals.

The results, however, vary. First of all, the quality of the reproduction depends greatly on that how well the head used in recording matches the head of the listener as otherwise the "coding" produced by the head is different and may produce weird results. Also, the quality is affected by the headphones used for reproduction. Finally, to reproduce the sound correctly, the sound pressure would have to measured on the tympanic membrane and the reproduced so that the sound pressure is exactly reproduced at the tympanic membrane. However, this is not yet possible.

Still, when everything matches well, the reproduction can preserve the spatial information and timbre quite well. Still, there exists one problem with this system. The binaural record-

ing is done with the head still in one position. This means that reproduction on works correctly if the listener also stays still as otherwise the conflict of moving head and the heard sound not positioning accordingly can reduce the perceived quality significantly.

5.3.2 Head-Related Transfer Function systems

Head-Related Transfer Function (HRTF) (Møller et al., 1995) systems are based on the same principle as the aforementioned binaural recording. The difference is that instead of recording using a head, the transfer functions of the head (artificial or real) are measured and used as filters to process recording of the original sound. The result can be exactly the as with binaural recording but using HRTFs has its advantages. As HRTFs are essentially just digital filters, it is possible to modify them or use them with different ways. One example is to track the head of the listener and use HRTFs accordingly so that the sound scene "stays still" even though the head is turned. This effectively removes the problem which was mentioned earlier in the binaural recording section.

5.3.3 Crosstalk cancelled stereo

Crosstalk cancelled stereo is a special application of binaural signals. Instead of using headphones for reproduction, two loudspeakers are used. The problem is that the two signals in binaural signals are meant to be divided strictly: one for the left ear and one for the right ear. However, with distant loudspeakers, these signals will mix and the result is not the intended one. This unwanted mixing of signals is called crosstalk. It is possible to remove it and this process is called crosstalk cancellation (Kirkeby et al., 1998). In this method, the crosstalk from the right loudspeaker to the left ear is cancelled by sending a inverted signal with a exact amount of delay from the left loudspeaker. Of course this inverted would then cause crosstalk to the right ear so it must be again cancelled. Using attenuating canceling signals, it is possible to dampen almost all crosstalk. Crosstalk from the left loudspeaker to the right ear is, of course, cancelled with the same scheme.

The result is on par with aforementioned binaural technologies when the listener is at the sweet spot. The problem is that this reproduction scheme only works well at the small sweet spot.

5.3.4 Spatial sound reproduction with multichannel loudspeaker systems

Multichannel loudspeaker systems use, just like the name says, multiple loudspeakers to reproduce the sound. Generally, this term has been used for systems using more than two loudspeakers. The general advantage of using more loudspeakers is that it allows the spatial information to be reproduced more faithfully.

Multiple loudspeakers are often used in a standardized layout. One of these (and the most popular in domestic use) is 5.1 which utilizes five surrounding loudspeakers and one (.1) low frequency loudspeaker. In this case three loudspeakers are at front with azimuths -30^o, 0^o and 30^o and two "surround" loudspeakers are at the sides with azimuths of -110^o and 110^o. Other popular layouts include 6.1 (addition of center back loudspeaker), 7.1 (two back surround speakers more) and in movie theaters, 10.2.

Systems utilizing the standardized layouts

Commercially most successful have been the surround sound systems which employ the aforementioned standardized loudspeaker layouts. Dolby, DTS and Sony are the most prominent developers in this area and provide proprietary (and widely used) formats for multichannel loudspeaker layouts.

The general idea is to record or mix the sound so that it uses the surrounding loudspeakers to produce an enveloping reproduction of the recording. In terms of quality, the spatial information is transmitted much better when compared to the stereo. However, the loudspeakers have to be positioned correctly for the reproduction to work well.

Ambisonics

Ambisonics (Gerzon, 1985) takes a bit different approach to the sound reproduction scheme. It aims to capture the complete sound field at a single position with a highly directional microphone array and then to reconstruct that sound field with a two- or three-dimensional array of loudspeakers. The theory of this method is based on the Huygens' principle (Longhurst, 1967). The resulting reproduction is theoretically at the sweet spot high quality as the sound field is reproduced. However, the problem is that the signals for loudspeakers are coherent when compared to each other. This produces comb-filter effects and localization of sound to the nearest loudspeaker due to the precedence effect when listened to in off

sweet spot. When combined with the fact that the sweet spot is relatively small, this can degrade the perceived quality of the reproduction.

Wavefield synthesis

Wavefield synthesis (AES Staff, 2004) is another approach to reconstruct the sound field. Instead of a single position, which was the aim of Ambisonics, wavefield synthesis aims to reproduce the sound field for a larger area which is usually the whole listening room. This is again based on the Huygens' principle (Longhurst, 1967) and done with large amount of carefully equalized loudspeakers in an array. The quality of the reproduction is theoretically good and offers an unique ability of creating virtual sound sources inside the room. The problem is that the loudspeakers need to be closely spaced for a good reproduction of high frequencies. Also, the recording and transmission of sound for wavefield synthesis is a complex task as there are quite a lot of separate sound channels used.

Vector Base Amplitude Panning

Vector Base Amplitude Panning (VBAP) (Pulkki, 1997) is a method for positioning virtual sources. It is a generalization of panning for any two- or three-dimensional loudspeaker setups. This panning can be applied to synthetic sounds or recorded sounds and produces as sharp sound source localization as possible with current loudspeaker configuration and amplitude panning methods. If multiple virtual sources are produced with VBAP, then it is possible to simulate some recorded situation like for example a classical orchestra.

Three-dimensional panning is done by presuming that all used loudspeakers are on a sphere centered at the listening point. If this is not the case, then delay and amplitude compensations have to be applied. This sphere is then divided to non-overlapping triangles with a loudspeaker in each corner of the triangle. The three loudspeakers in the triangle are then used to produce virtual sound sources which are located in the area of the triangle. This is done using equation

$$\vec{p} = \mathbf{L}\vec{g} = \begin{bmatrix} \vec{l}_1 & \vec{l}_2 & \vec{l}_3 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}. \tag{5.1}$$

Here, $\vec{p}$ is the unit vector pointing towards the virtual source, $\vec{l}_n$ are the unit vectors pointing

to the loudspeakers of the triangle and g_n are the corresponding gains of the loudspeakers. To solve the gains for a virtual source positions, matrix $\mathbf{L}$ is inverted and equation

$$\vec{g} = \mathbf{L}^{-1}\vec{p} \tag{5.2}$$

will be received.

These gains are then normalized with equation

$$\sum_i g_i^2 = 1 \tag{5.3}$$

so that constant sum of energy is achieved independent of direction.

The two-dimensional case differs in that instead of a sphere, the loudspeakers are on a circle and non-overlapping sets of two loudspeakers are used. These reduces the equation 5.1 to form

$$\vec{p} = \mathbf{L}\vec{g} = \begin{bmatrix} \vec{l_1} & \vec{l_2} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}. \tag{5.4}$$

Chapter 6

Directional Audio Coding

Directional Audio Coding (DirAC) is a recently proposed method for spatial sound reproduction (Pulkki, 2007). It offers a method of analyzing and synthesizing spatial audio based on psychoacoustic assumptions. Basic theory was founded in the SIRR (Spatial Impulse Response Rendering) research (Merimaa and Pulkki, 2005) and Directional Audio Coding is essentially a further developed version of it which is capable of processing any audio signal instead of impulse responses.

In this chapter, there will first be a basic explanation of the DirAC algorithm. Then, different parts will be explained separately. Finally, at the end of the chapter, new propositions to the algorithm will be presented.

6.1 Basic idea

Directional Audio Coding algorithm is divided to two parts, analysis and synthesis. Analysis part analyzes a B-format signal (or any other signal which can be converted to B-format) in frequency bands for the direction of arrival of the sound and the diffuseness of the sound. This information can then be transmitted as meta-data with the audio signal. In synthesis transmitted meta-data is used as parameters to synthesize separately the non-diffuse sound and the diffuse sound which are then combined to produce an image of the original spatial sound.

This algorithm has been developed using a few fundamental psychoacoustic assumptions (Pulkki, 2007).

- The direction of arrival of the sound transforms into ILD, ITD and monaural localization cues.
- The diffuseness of sound transforms into interaural coherence cues.
- Timbre depends on the monaural spectrum and the ITD, ILD and interaural coherence cues.
- The direction of arrival, diffuseness and spectrum of the sound measured in one position with the temporal and spectral resolution of human hearing determines the auditory spatial image the listener perceives

The algorithms described in this chapter are directly based on these assumptions. The key idea is that if the cues presented in the first three assumptions are reproduced faithfully with the temporal and spectral resolution of human hearing then the perceived spatial image is similar to the original scenario.

6.2 B-format signal

As the B-format signal is the chosen form for audio in DirAC, it is useful to describe it well. B-format is the natural signal format produced by a soundfield microphone and used extensively in Ambisonics (see chapter 5.3.4). There are different orders of B-format signals but the most common one (and the one used in DirAC) is the first order B-format constituting of four channels. These channels are produced with microphones which in order are: an omnidirectional microphone producing signal W and signals from three figure-of-eight microphones pointing towards front, left and up producing signals X, Y and Z correspondingly. Usually, the W signal is scaled with $\frac{1}{\sqrt{2}}$.

6.3 Analysis

DirAC analysis is performed to estimate the direction of arrival and the diffuseness of a sound signal. These values can be estimated from the intensity and the energy of the sound signal. The estimation process can be performed equivalently in time domain or frequency domain. As this thesis concentrates on the STFT approach it is logical to present the frequency domain versions of the equations here.

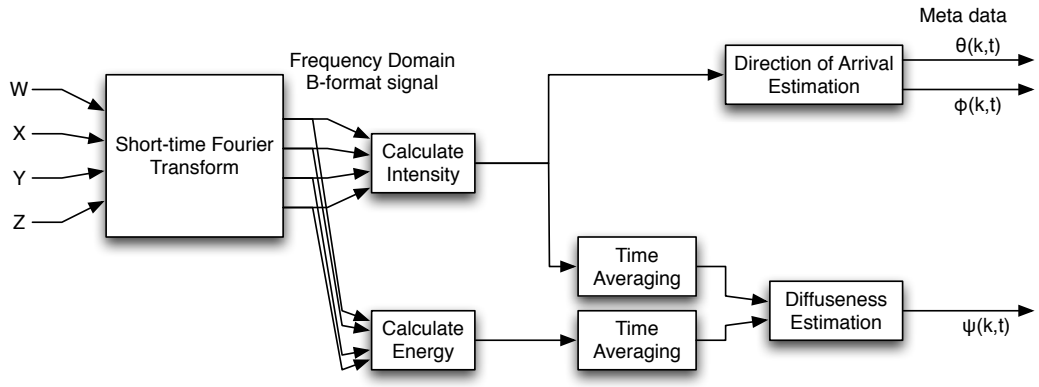


Figure 6.1: Block diagram of a STFT-based DirAC analysis.

A block diagram of the DirAC analysis can be seen in Fig. 6.1. This section will move through the diagram from left to right. STFT DirAC analysis begins with dividing the B-format signal to time blocks and transforming them to frequency domain. After this, the actual analysis process begins. Direction of arrival and diffuseness are estimated based on energetic analysis. The theory was already established in Merimaa and Pulkki (2005) and Pulkki and Merimaa (2006) and finalized for DirAC in Pulkki (2007).

To do the estimation, first, it is mandatory to estimate the instantaneous active intensity and energy. Like it was presented in chapter 2.1, intensity and energy can be calculated from sound pressure and particle velocity (equations 2.1 and 2.2). With B-format signal, the omnidirectional W signal can be used as a pressure estimate and the figure-of-eight signals X, Y and Z can be converted to directional particle velocity estimates. Formally, this can be put to equations (equations 6.1, 6.2 and 6.3).

$$P(k) = W(k) \tag{6.1}$$

$$\vec{X}'(k) = X(k)\hat{e}_x + Y(k)\hat{e}_y + Z(k)\hat{e}_z \tag{6.2}$$

$$\vec{U}(k) = \frac{1}{\sqrt{2}Z_0} X'(k) \tag{6.3}$$

Here $\hat{e}_x$ denotes an unit vector to the corresponding direction. k is used to denote the frequency band here. Signals X, Y and Z are combined to a single vector signal $\vec{X}'$ to

simplify further equations.

If these equations are placed in the aforementioned intensity and energy equations we get equations 6.4 and 6.5.

$$\vec{I}(k) = \frac{1}{\sqrt{2}Z_0}\Re\left\{W(k)^*\vec{X}'(k)\right\} \tag{6.4}$$

$$E(k) = \frac{\rho_0}{2Z_0^2}\left[|W(k)|^2 + \frac{|\vec{X}'(k)|^2}{2}\right] \tag{6.5}$$

Note that only the active intensity is desired and thus the real part is used.

Now it is possible to do the estimation for the desired variables. The direction of arrival can be estimated from the active intensity vector. This is done by assuming that the sound comes from the exactly opposite direction compared to the active intensity vector. DirAC uses azimuth θ and elevation φ to define the direction of arrival so to estimate these angles from the active intensity vector, equations 6.6 and 6.7 will be used.

$$\theta(k) = \tan^{-1}\left[\frac{-I_y(k)}{-I_x(k)}\right] \tag{6.6}$$

$$\varphi(k) = \tan^{-1}\left[\frac{-I_z(k)}{\sqrt{I_x^2(k) + I_y^2(k)}}\right] \tag{6.7}$$

I_x here is the magnitude of the x-component of the intensity vector which is formally the dot product $I_x = \vec{I} \cdot \hat{e}_x$. Other components are calculated similarly.

Diffuseness can be estimated from the intensity and the energy of the signal like it was presented in Eq. 2.3. By substituting equations 6.4 and 6.5 to that, the diffuseness equation

$$\psi(k) = 1 - \frac{\sqrt{2}\|\Re\{W^*(k)\vec{X}'(k)\}\|}{|W(k)|^2 + \frac{|\vec{X}'(k)|^2}{2}} \tag{6.8}$$

for STFT DirAC will be formed. This method was already presented in Merimaa and Pulkki (2005) and is based on the amount of energy transfer. However, another method has been

proposed for estimating the diffuseness based on only the intensity estimate (Ahonen et al., 2009). The formula can be seen in equation

$$\psi(k) = \sqrt{1 - \frac{\|\langle \vec{I}(k) \rangle\|}{\langle \|\vec{I}(k)\| \rangle}}. \tag{6.9}$$

This method is based on the idea of calculating the ratio between the magnitude of time averaged intensity and time averaged magnitude of the same intensity.

Note from the block diagram (Fig. 6.1) that independent of the diffuseness estimation method, some form of time averaging will be applied. This time averaging is an important parameter and can change the behaviour of the analysis quite a lot.

6.4 Synthesis

DirAC synthesis can be divided to three major parts: virtual microphone generation, non-diffuse sound synthesis and diffuse sound synthesis. Depending on the implementation the structure of these blocks may differ but the general structure stays intact. The block diagram of a generic STFT version of DirAC synthesis can be seen in Fig. 6.2 where the three separate blocks can be clearly seen.

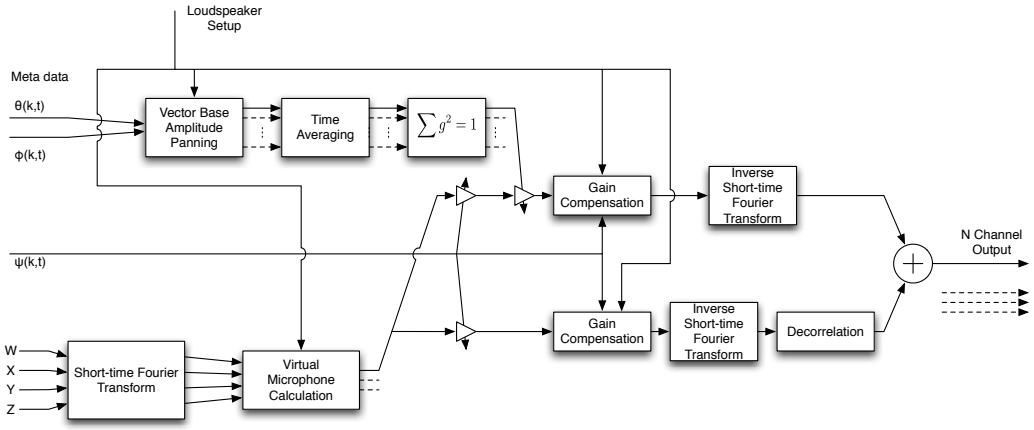


Figure 6.2: Block diagram of a STFT-based DirAC synthesis.

The processing starts with the virtual microphone generation where appropriate signals for each loudspeaker are created. Then the signal is input to non-diffuse sound and diffuse

sound synthesis parts. These algorithms apply different synthesis methods to the signal. The results are then mixed together in frequency bands with mix ratios defined by the estimated diffuseness of the sound received from the analysis. Notable fact is that with real-life sound there is always some reverberation in the room (with the exception of anechoic chambers and large outdoor spaces) and thus the signal is not usually purely non-diffuse. Also, it is extremely rare that the signal would be purely diffuse as this would demand sound coming evenly from all directions.

6.4.1 Virtual microphones

STFT DirAC transmits signals as frequency domain B-format frames which contains the frequency domain representation of a time frame for each B-format channel. In synthesis, it is needed that this B-format signal is converted to virtual microphone signals which correspond to the loudspeaker setup used for reproduction. These virtual microphone signals can then be used in both non-diffuse sound and diffuse sound synthesis algorithms.

The simplest choice of creating required virtual microphones from the B-format signal is to use omnidirectional virtual microphones. This means that the pressure signal W in B-format is directly replicated to all loudspeakers. However, omnidirectional microphone patterns have disadvantages which were already presented in Pulkki (2007). Firstly, all signals will be fully correlated and this in turn creates comb-filtering effects and need for better decorrelators (see section 6.4.3) in diffuse sound synthesis.

Secondly, if two separate sound sources share frequency bands (for example two male humans speaking in different directions) then the estimated direction will vary between them on those frequency bands. As there is a need for time averaging loudspeaker gains (see section 6.4.2), this leads to a problem because the virtual microphone signals are in this case fully correlated. The result is that the separate sources appear to be closer to each other than they were in the original situation. These two fallacies render the use of omnidirectional virtual microphones practically useless for high quality implementations.

If there are more than the pressure signal present in the transmitted B-format signal then it is possible to use directional virtual microphones directed towards the loudspeakers. Corresponding signals can be calculated simply from the B-format signal with a linear combination using equation

$$S_n(k) = \frac{2-\kappa}{2}W(k) + \frac{\kappa}{2\sqrt{2}}\left[\cos(\theta_n)\cos(\varphi_n)X(k) + \sin(\theta_n)\cos(\varphi_n)Y(k) + \sin(\varphi_n)Z(k)\right]. \tag{6.10}$$

Here S_n is the virtual microphone signal, θ_n and φ_n are the azimuth and elevation of the n^{th} loudspeaker and $0 \leq \kappa \leq 2$ represents the directional pattern of the microphone. Value 0 of the κ corresponds to an omnidirectional microphone pattern, value 1 corresponds to a cardioid pattern and value 2 corresponds to a dipole pattern. As it can be seen, previously mentioned omnidirectional microphones are only a special case of this formula.

Previously it was informally found that a pattern between dipole and hypercardioid seems to produce best perceptual results (Vilkamo et al., 2008). However, there might be a need to decide the correct pattern on a case to case basis. Note that omnidirectional pattern has the advantage that only one audio signal needs to be transmitted instead of four (or three in the case of dipoles) which results in significant data reduction.

6.4.2 Non-diffuse sound synthesis

The non-diffuse sound synthesis of the DirAC aims to synthesize the parts of the sound signal which come from only one direction or more exactly the non-diffuse sound. As with all DirAC processing, this is done to each frequency band separately. Essentially the whole process is only an application of different zero-phase FIR filters for each loudspeaker channel for the current time moment. These filters can be calculated with a single equation (Eq. 6.11) but it can be divided to three separate functional blocks which will be described further here.

$$g_{\mathsf{non\_diffuse}}(k,n) = \frac{g_{\mathsf{vbap}}\left[n, \theta(k), \varphi(k)\right]\sqrt{1-\psi(k)}}{\sqrt{\left[1-\psi(k)\right] + \psi(k)g_{\oint}^2}} \tag{6.11}$$

The first component applied to the signal is Vector Base Amplitude Panning (VBAP) which was described in section 5.3.4. VBAP is used to position the sound signal to the analyzed direction which is stored in the metadata. This applies correct gain values for each frequency band and loudspeaker channel combination. Essentially this means that a filter of certain form is applied to each channel (see Fig. 6.3). However, if the gain values are applied directly, the result is not always pleasant. This is because in some cases the analyzed direction varies rapidly through time and frequency. This creates audible noisy artifacts.

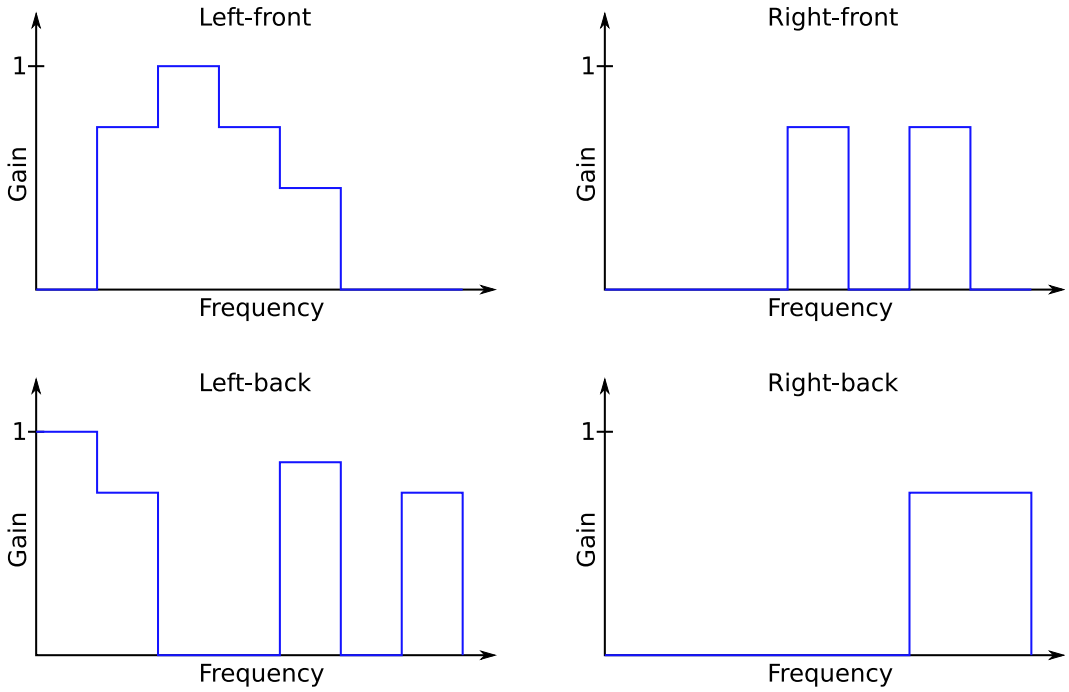


Figure 6.3: An example of channel specific VBAP gain filters for a loudspeaker setup consisting of four loudspeakers on the horizontal plane. Six frequency bands are used in this case. Notice that the squared sum of gains for each frequency band should be one.

To reduce this problem, a lowpass filter is applied which smoothes the transitions between consecutive gain values in time. This filter can be either FIR or IIR but previous implementations used mainly symmetric FIR filters. Time constant of the filter has to be selected carefully so that it works well for all audio signals. That means that the constant should be small enough to not to introduce any sluggishness to the sound and at the same time large enough to remove audible artifacts. Additionally, the filtering can change the balance of the gain coefficients between channels so it is necessary to normalize the gain coefficients so that the squared sum of the channel gains is constantly one. This can be achieved with equation

$$g_{\text{vbap\_norm}}(k,n) = \frac{g_{\text{vbap}}(k,n)}{\sqrt{\sum_{m=1}^{N} g(k,m)}}. \tag{6.12}$$

Here $g(k,n)$ is the gain coefficient for frequency band k and loudspeaker channel n. N is the amount of loudspeaker channels. The result is the desired g_{vbap} term in the Eq. 6.11.

The second component contains a gain compensation needed due to the directionality of the virtual microphones. This is needed because by using directional microphone patterns, some of the energy is lost for each loudspeaker. This loss has to be compensated so that the overall sound pressure level does not depend on the chosen microphone pattern. Vilkamo (2008) devised a robust gain compensation method which considers the total energy loss of all loudspeakers and uses that for compensation. The result is formulated in equation

$$g_{\mathsf{nd\_mic\_compensation}}(\psi(k)) = \frac{1}{\sqrt{[1-\psi(k)] + \psi(k) g_{\oint}^2}} \tag{6.13}$$

for non-diffuse signals. Here ψ is the diffuseness of the frequency band k and $g_{\oint}$ is the microphone gain for the diffuse field defined as

$$g_{\oint}(\kappa) = \sqrt{1 - \kappa + \frac{\kappa^2}{3}} \tag{6.14}$$

which depends on the directional pattern parameter κ of the virtual microphones used.

The third and final component handles mixing between non-diffuse and diffuse signals based on the analyzed diffuseness. For non-diffuse sounds the filter created is defined with equation

$$g_{\mathsf{nd\_mix}}(\psi(k)) = \sqrt{1 - \psi(k)}. \tag{6.15}$$

The combination of these three components produces the non-diffuse sound synthesis part of the DirAC. This whole process can be illustrated by studying the magnitude responses of the filters they produce and how they combine. This can be seen in Fig. 6.4 where microphone compensation is presumed to be one (that is, omnidirectional pattern is used) for simplicity.

6.4.3 Diffuse sound synthesis

The diffuse sound synthesis part of DirAC complements the non-diffuse sound synthesis and synthesizes sounds which are not part of the analyzed directional sound. Similarly to the non-diffuse sound synthesis, the diffuse sound synthesis can be divided to separate com-

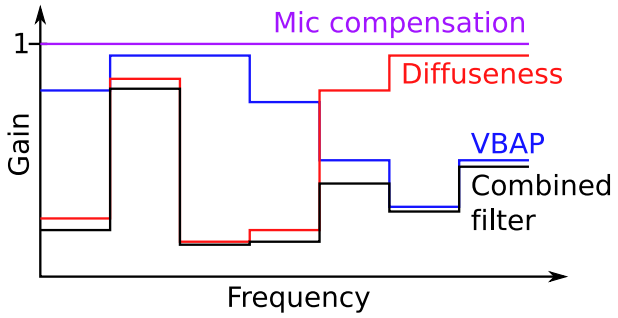


Figure 6.4: Separate synthesis filters which are then combined to one synthesis filter.

ponents, four in this case. Also, a similar gain formula for diffuse sound can be formulated (Eq. 6.16) which contains three of the four components.

$$g_{\text{diffuse}}(k,n) = \frac{1}{g_{\oint}} \sqrt{\frac{A(n)}{E[A(n)]}} \sqrt{\frac{\psi}{N}} \tag{6.16}$$

The first and second components apply gain compensations to the signal which are dependent on the used parameters. The first component takes account the used loudspeaker setup. To produce a diffuse sound, it is necessary that the sound arriving from all directions is equally strong. With symmetrical loudspeaker setups this happens naturally but with asymmetrical setups there is a need to compensate the gain differences between different directions. This can be achieved by normalizing the energy of each loudspeaker with the area which it is covering. The needed compensation for a loudspeaker can be seen in equation

$$g_{\text{d\_area\_compensation}}(n) = \sqrt{\frac{A(n)}{E[A(n)]}}. \tag{6.17}$$

Here n is the number of loudspeaker channel and $A(n)$ is a function which calculates the area containing all points on a listening point centered sphere which are closest to the loudspeaker n. E defines expectation which in this case transforms to an average of the all areas calculated. This equation diminishes those loudspeakers which are close to each other and strengthens those which are wide apart.

The second component applies a gain compensation due to the directional pattern of the used virtual microphones. This is quite similar to the second component of the non-diffuse sound synthesis and can be formulated to

$$g_{\mathsf{d\_mic\_compensation}}(\kappa) = \frac{1}{g_{\oint}(\kappa)}. \tag{6.18}$$

The diffuse microphone gain $g_{\oint}(\kappa)$ found here was already defined in Eq. 6.14.

The third component is also similar to the third component of the non-diffuse sound synthesis and handles mixing of output as per the estimated diffuseness value. Additionally it normalizes the gain with the number of the loudspeakers used. This can be seen in equation

$$g_{\mathsf{d\_mix}}(\psi(k)) = \sqrt{\frac{\psi}{N}}. \tag{6.19}$$

Finally, the fourth component is decorrelation which is needed in synthesizing diffuse sound field from a smaller amount of signals. This will be described in the next section more thoroughly.

6.4.4 Decorrelation

Decorrelation in DirAC is used to reduce coherence between different loudspeaker signals in diffuse synthesis. To be more exact, the decorrelation method used here is properly called audio signal decorrelation. The difference to a classical signal decorrelation, where signals are made orthogonal to each other, is that audio signal decorrelation has an additional restriction that the timbre produced by the sound signal should not change.

Audio signal decorrelation is usually performed with separate filters which are designed according to the restrictions. Timbre of the sound signal is preserved quite well if the magnitude response of the filter is kept at a constant value at all frequencies. Thus, decorrelation filters are ideally all-pass filters. That also means that the modifications are performed to the phase spectrum of the signals. As human hearing is somewhat insensitive towards phase it is possible that even drastic modifications can be made without any other audible artifacts than the decrease of coherence which is heard as increase of sound spatiality.

There exists quite many different audio signal decorrelators. All current methods make some compromises to achieve decent decorrelation and there does not yet exist a perfect audio signal decorrelator which would work perfectly for all signal types. The following subsections will describe different methods used currently and analyze their qualities and expected performance.

Critical bands with random time shifts

This decorrelation method was proposed by Bouéri and Kyriakakis (2004). The general idea is to first divide the input signal to separate preferably non-overlapping frequency bands based on the critical bands of the human hearing. Then a different constant delay is randomized for each desired output channel. Boundaries for the delay amount are set so that the delay must be large enough to reduce coherent summation between the original and processed signals and the delay must be small enough to keep the timbre of a single channel unchanged (under 20 ms). In addition, the delays of adjacent decorrelators have to be set so that their time shifts (that is, delays) are correctly aligned to ensure that there is no destructive summation at the boundary region. More precisely,

$$S_h - S_l = kT_{lim} \quad , k \in \mathbb{Z} \tag{6.20}$$

has to hold for all band boundaries. Here S_h is the time shift applied to the higher frequency band and S_l is the time shift applied to the lower frequency band, T_{lim} is the period of the boundary frequency dividing the two frequency bands. This procedure does not completely remove destructive summation as with realistic band-pass filters the boundary is not one discrete point and the change from one band to the next happens "slowly".

Additionally, the maximum applicable delay has to be made frequency-dependent as otherwise on high frequencies there would be audible artifacts due to the large time shifts compared to the period of the signal. Bouéri and Kyriakakis (2004) proposed as one solution to make the maximum allowable time shift for the frequency band dependent to the length of the period of the largest frequency on that same frequency band. However, Vilkamo et al. (2008) and Laitinen (2008) informally found that the use of this frequency dependency does not decorrelate the signal enough on high frequencies.

The compromise done here is between the amount of perceived decorrelation and phase shift and echo artifacts. It is notable that if the input signal is complex enough then the artifacts tend to be inaudible. Also, impulsive sounds tend to produce artifacts when a long filter is applied to them. Note that for real-time processing, the filter has to be causal and thus have to be delayed accordingly.

Convolution with noise bursts

This method was studied in paper by Hawksford and Harris (2002). It was also one of the proposed methods in SIRR ((Merimaa and Pulkki, 2005) and (Pulkki and Merimaa, 2006)). The general idea is quite simple in this method. A different white noise burst is generated for each desired output channel to act as a decorrelation filter. This filter is then equalized as the magnitude response is not constant even though white noise would imply that[1]. Finally, these noise bursts are then convolved with the input signal to receive the multichannel output signal.

The noise burst can be created with a multitude of methods. The simplest method is to use rectangular window which simply means taking directly a number of values from a random number generator. This decorrelates signal but also generates noise-like artifacts to the signal, especially after impulsive sounds. A better choice is to use an exponentially decaying window. Due to the exponentially decreasing shape of the time envelope, the created artifact is less noticeable. However, the noise burst needs to be longer than with rectangular window for low frequencies to decorrelate. Even more refined method is to use exponentially decaying window where the decay rate is frequency dependent. This means that low frequencies have slower decay rate to produce better decorrelation and high frequencies have faster decay rate so that there is less artifacts with impulsive sounds which contain a lot of high frequency components.

This decorrelation method is a good choice when the input signal is highly reverberant as the artifacts created from convolution with noise only add to the perceived diffuseness of the sound. However, impulsive and dry sounds suffer from temporal smearing when this method is applied. Additionally, to decorrelate low frequencies correctly, the length of the filter has to be quite long which might not be desirable in real-time processing.

Time-varying phase methods

This group of methods include few different solutions to produce decorrelation. The common thing is that the all-pass filters used in these methods are time variant. This theoretically can produce better decorrelation as the time varying nature in itself can be harnessed for decorrelation purposes. In simple terms, these filters are basically similar to aforementioned delay inducing filters. The difference is that now the delay is changing through time.

[1] This is due to the fact that white noise has only statistical magnitude response of one. Thus it would need an infinite filter length for a constant magnitude response.

However, the problem is that time varying filters also create another kind of artifacts to the sound.

The general idea of these methods is to directly modify the phase response of the signal to generate the output signals. This modification can be done in time domain with all-pass filters or efficiently in frequency domain by directly modifying the phase response. Here, only the frequency domain methods will be discussed as they are simple and efficient to implement for real-time applications. The simplest and most brutal solution to apply time variant phase modification is to randomize the phase for each input block. This method was proposed as a one decorrelation method in the SIRR (Pulkki and Merimaa, 2006) and produced reasonable results there. However, general audio signal is quite different from an impulse response and changing the whole phase response rapidly will generate quite serious artifacts to the output signal.

A more subtler method was patented by Herre and Buchner (2007). In this method the phase of the input signal is modulated with a continuous waveform. For each desired output signal a different frequency of the modulator is selected. Additionally, the amplitude of the modulator is made frequency dependent so that on low frequencies, where human ear is more sensitive to the phase, phase modification is smaller. On the other hand, on high frequencies the phase modification can be quite drastic without any audible artifacts. This means that the parameters should be heuristically tuned so that almost all artifacts disappear. However, this method tends to have some intrinsic flanging[2] in the output. The good side is that this method can be applied with short FFT window sizes and still produces reasonable decorrelation. Also, this method is still quite efficient to implement. This means that this method suits really well for real-time applications.

6.5 New propositions

This section describes ideas which have not been studied in the context of DirAC yet and have been implemented for this thesis.

[2]Flanging here means the effect generated by a flanger effect which sounds a bit like the sound is vibrating rapidly in space.

6.5.1 Multi-resolution STFT

Multi-resolution STFT is defined here to be a method where the signal is divided to frequency bands (generally non-overlapping) and each band is processed with a different window size. This enables the use of longer windows on low frequencies to get better frequency resolution. On high frequencies lower frequency resolution is needed but instead transients need accurate time resolution which can be achieved with a smaller window.

Multi-resolution STFT (MRSTFT) is not a new invention. It has been used for example in signal analysis (Smith, 2008c). The advantage in the case of DirAC is the better adaptation to the resolution properties of the human hearing. However, multiple resolutions with perfect reconstruction is also multiple times more complex to compute as all computations have to be performed for each resolution. Additionally, a bandpass filter has to be applied to separate the bands where different window sizes are used. However, if the perfect reconstruction property is relaxed somewhat and band limits are selected appropriately then band-pass filters can be designed as "ideal" in frequency domain. Put simply, this means that only the values inside the pass-band are calculated for each band and other values are simply discarded. The result contains some artifacts but they are generally inaudible. The number of computations needed for this non-perfect method is closer to the amount that a normal single resolution STFT needs.

Using multiple resolutions creates some requirements for the DirAC parameters. All averaging filters have to be tuned for all window sizes correctly. Obviously they should be related to each other and to the window size. This is further discussed in chapter 7.2.3. Also, the dividing boundary frequencies for the bandpass filters have to be selected correctly to receive optimal results.

6.5.2 Bin-based processing

The previous real-time STFT DirAC implementations were designed to work in spatial sound transmission scheme and thus used data reduction in processing. That means that even though the processing was done in frequency domain the estimates were combined so that there is only one value set for each frequency band. However, there is no need for data reduction in a high quality implementation as the data is not transmitted.

Pure bin-based processing should produce good results but it might be useful to apply frequency-dependent smoothing through frequency for values. This is done to mimic the

critical band property of the human hearing (see chapter 3.2.1). This smoothing will be done to the intensity vectors in analysis and separate methods can be used for direction and diffuseness estimation. An educated choice for these methods is to use ERB for the bandwidth of the smoothing window.

A block diagram of MRSTFT-based DirAC analysis with added frequency smoothing can be seen in Fig. 6.5. DirAC synthesis does not change significantly as STFT blocks are only replaced with multi-resolution variants.

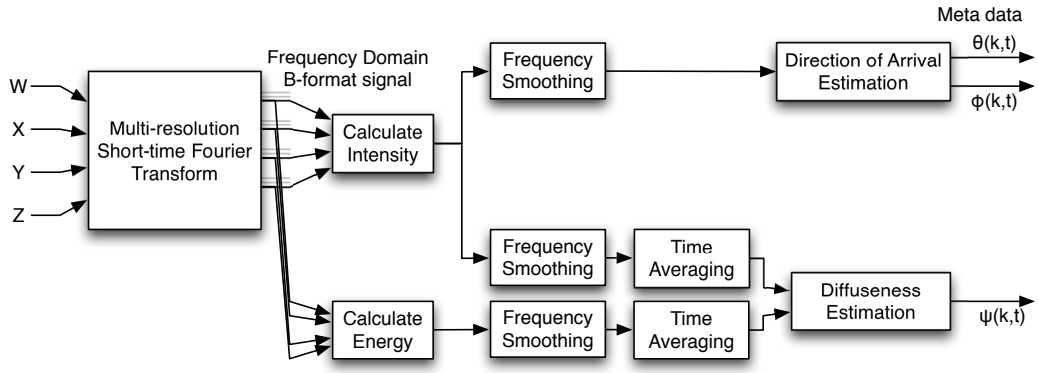


Figure 6.5: A block diagram of a MRSTFT-based DirAC analysis with frequency smoothing.

6.5.3 Hybrid Decorrelation

Previous DirAC implementations (Pulkki, 2007; Laitinen, 2008; Vilkamo et al., 2008) presented and tested different decorrelation methods and noted few observations in informal tests.

- Channel and frequency-band dependent time invariant delays produced generally best results. However, at high frequencies longer delays were needed due to the properties of directional human hearing (see 3.2.3)
- Exponentially decaying noise bursts perform well on high frequencies where they decorrelate the signal strongly. However, on low frequencies the length of the burst needs to be long enough so that the noise is statistically random and it can perform as a decorrelator. This in turn leads to audible noise bursts when transient sounds are processed.

In the Pulkki and Merimaa (2006) there was a similar case where different decorrelation schemes performed better at different frequency bands. In the end, a hybrid method was devised which applied the best scheme for each frequency band. Similarly, it is prudent to assume here that a combination of two methods might produce all-round better performance.

The two chosen decorrelation methods for the hybrid method are the previously mentioned delay-based decorrelation method and exponentially decaying noise burst -based method. Delays will be used for low frequencies and noise bursts will be used for high frequencies. Dividing frequency will be around 1500 Hz which is the dividing boundary in human hearing deciding if ITD or ILD is the decisive cue for direction detection (Feddersen et al., 1957).

Chapter 7

Implementation

This chapter describes in depth the actual implementation of DirAC made for this thesis. First, the design principles will be described. Then, all the design choices are studied and problems caused by them are analyzed.

7.1 Design principles

When creating the DirAC implementation, there were three main guidelines which were followed. The first and most important one was that the system should produce high quality results. This means that audio processed with the created DirAC implementation should be comparable or better in perceived quality when compared with previous implementations. Also, difference to a reference scenario should be small.

The second guideline was that the produced implementation should be easily portable, should constitute of separate blocks and should be changeable. This is done to ensure future development of the technology and to offer a possibility to add new parts to the system.

The final guideline was optimality. That means that the system should use as few resources as possible without breaking the first and the second guidelines.

7.2 Design choices

This section describes design choices made on the algorithmic level for this implementation. Properties of the produced software package will be described in chapter 9.

7.2.1 Overlap-add

The overlap-add method was used for calculating real-time filtering in DirAC. This method was already well described in section 4.4.2 so only chosen parameters will be described here. The chosen window was selected to be the Hann window which is a raised cosine window. This window type is widely used and offers good properties for general purpose signal processing. Additionally, it produces constant overlap-add property when the overlap is selected to be 50% of the window length and the window is of periodic form[1].

7.2.2 Multi-resolution STFT

Multi-resolution STFT itself has two parameters which have to be selected wisely to produce good results. These parameters are the different resolutions and the frequency bands assigned to them. There are various means of finding a somewhat optimal solution for them. Generally, the chosen parameters should somehow relate to the properties of human hearing.

One method is to use rules derived from the properties of hearing. First, the number of desired sizes is selected. Then, the STFT block size and corresponding frequency band are selected so that the resulting time resolution is always better than that of the human hearing and the bandwidth of a frequency bin is smaller than the ERB bandwidth on that frequency band. As an additional requirement, band limits must be matched together to ensure that there are no gaps or overlaps in between. The result produces in some sense optimal division regarding to the human hearing.

Another method is to subjectively evaluate different reasonable parameter values using signals which are challenging to reproduce otherwise. An example of a challenging signal is applauds as they contain lot of impulsive sounds which would cause time smearing with a large block size. Using this method can be an arduous task as often the other DirAC

[1] This is achieved by creating the window with the desired window length plus one and then removing a zero from one end.

parameters have to be tuned for each resolution separately. However, theoretically through meticulous listening tests it would possible to create an optimal parameter set with this method.

A third method is to optimize speed. This means that first, parameters are optimized with any other method to desired setup and then the results are tuned so that multirate signal processing can be applied. This is especially good choice if the implementation is aimed for a hardware solution and it is advantageous to have only one size of FFT to be used. Thus, the parameters are tuned so that different frequency bands can be downsampled to the same FFT size which is used by the highest frequency band. This method leads to simplifications in the hardware architecture and thus reductions in the build costs.

All in all, better results compared to a single resolution STFT DirAC should be received by just using the general rule of using large block size on the low frequencies and small block size on the high frequencies and selecting the band limits with an enlightened choice.

7.2.3 Time averaging

Selection of time averaging parameters of DirAC is a more complex task when MRSTFT is used. For simplification, the time averaging filters used in the implementation were one pole IIR filters. This goes against previous suggestions (see chapter 6.4.2) of using average filter with a symmetrical impulse response. However, applying symmetrical time window for STFT already does some averaging and thus the resulting window is a more complex one instead of the one generated by a simple one pole filter.

Choosing the time constants for these one pole filters is not a simple task even with a single resolution, as the optimal values depend on the content of the processed signal. Generally, the values are tuned experimentally so that results are pleasant with most signals. However, for each time resolution, the parameters have to be set separately creating a lot of work. To simplify this, a formula was fitted to sets of parameters which were found to be good in informal listening of different sound samples. The result for loudspeaker gain averaging is

$$\tau_{\mathsf{gain}}(N) = \frac{8.3 \cdot 10^{-5}}{2.0^{-2\log_2(2048)}} 2.0^{-2\cdot\log_2(N)} \tag{7.1}$$

and for diffuse estimation averaging it is

$$\tau_{\mathsf{diffuse}}(N) = \frac{5.95 \cdot 10^{-5}}{2.0^{-2 \cdot \log_2(2048)}} 2.0^{-2 \cdot \log_2(N)} \tag{7.2}$$

Here N is the size of the FFT used and the reference constant is measured from a 2048 point FFT. These time constants define correct values for the lowest frequencies. To calculate the corresponding filter coefficient m from these time constant equation

$$m = 1 - \frac{1}{\tau f_s} \tag{7.3}$$

is used. However, for high quality, a frequency dependent scheme is needed. This is achieved with equation

$$m(k) = 1 - \frac{1}{\tau f_s} - \frac{k}{N \frac{1}{\tau f_s}} \tag{7.4}$$

which produces linear dependency from the frequency. k is the number of the frequency bin.

To further enhance control over frequency dependency, the maximum value of averaging is inserted to the formula and a tuning coefficient γ is introduced. This results in equation

$$m(k) = 1 - \frac{1}{\tau_{\mathsf{max}}(N) f_s} - \frac{k\gamma}{N \frac{1}{\tau_{\mathsf{max}}(N) f_s}}. \tag{7.5}$$

This can be solved for the corresponding time constant equation with the aid of Eq. 7.3. The result is equation

$$\tau(k) = \frac{1}{f_s \left(\frac{1}{\tau_{\mathsf{max}} f_s} + \frac{k\gamma}{N \frac{1}{\tau_{\mathsf{max}} f_s}} \right)}. \tag{7.6}$$

It was found that best results were received when γ is set so that at high frequencies there was almost no time averaging.

7.2.4 Synthesis filters

As noted in section 6.4.2, DirAC synthesis performs time-variant zero-phase filtering on the virtual microphone signals. This causes certain problems to the system. Firstly, zero-phase filtering requires that delay is added to the signal to produce proper causal output which can be overlap-added. Secondly, the original signal was zero padded to accommodate space for the convolution result. The problem is that the filter creation is done in frequency domain and corresponding to the zero padded signal. This means that the filter is actually longer than it is allowed as the correct length would be the length of the original data without zero-padding. This problem generated audible noisy distortion when the length of the FFT was short and bin-based processing was used.

The first problem can be solved easily in time domain as delay can be introduced with ring buffers. However, as the exact amount of delay needed is relative to the length of the calculated FFT and the filter is zero phase, it is simple to introduce this delay in the frequency domain. If it is assumed that the signal is zero padded to double length and the filter is real symmetric, then a correct amount of delay can be introduced by making the phase component of the filter frequency response to change by the angle of $-\frac{\pi}{2}$ between each frequency bin starting from 0. This introduces a delay of exactly a quarter of the FFT window to the filter which is the desired amount.

The second problem can be solved by inspecting the properties of zero padding in time domain as it is desired to make the filter length equal the signal length without zero padding. Zero padding in time domain applies optimal interpolation to the signal in frequency domain (Smith, 2008d). To create the same result in reverse it is obvious that interpolation has to be applied. This interpolation has to be done so that roughly half of the data is interpolated from the other half. This is done correctly when the source data contains frequency bins $1, 3, 5, ..., N-1$ and interpolated frequency bins are $2, 4, 6, ..., N$.

Piecewise fitted polynomial interpolation (with linear interpolation as a special case) offers an efficient algorithm for interpolating data so it was an obvious choice to apply it for this case. It was presumed that this method would remove a lot of unwanted artifacts from the sound. However, this was not the case and the change was inaudible. It seems that this interpolation method still leaves too much length to the impulse response of the filter.

Less efficient method of interpolation is to actually calculate IFFT of the filter and then window the impulse response to the correct length (Fig. 7.1). This essentially performs the similar processing to the synthesis filter which has been done to the signal which is to be

filtered. Obviously it is smart to use same window function here so that the filter response overlap-adds cleanly. Resulting windowed filter can now be transformed back to frequency domain with FFT and used for filtering the input signal. Resulting sound is artifact free as expected but this method increases the number of calculations needed considerably.

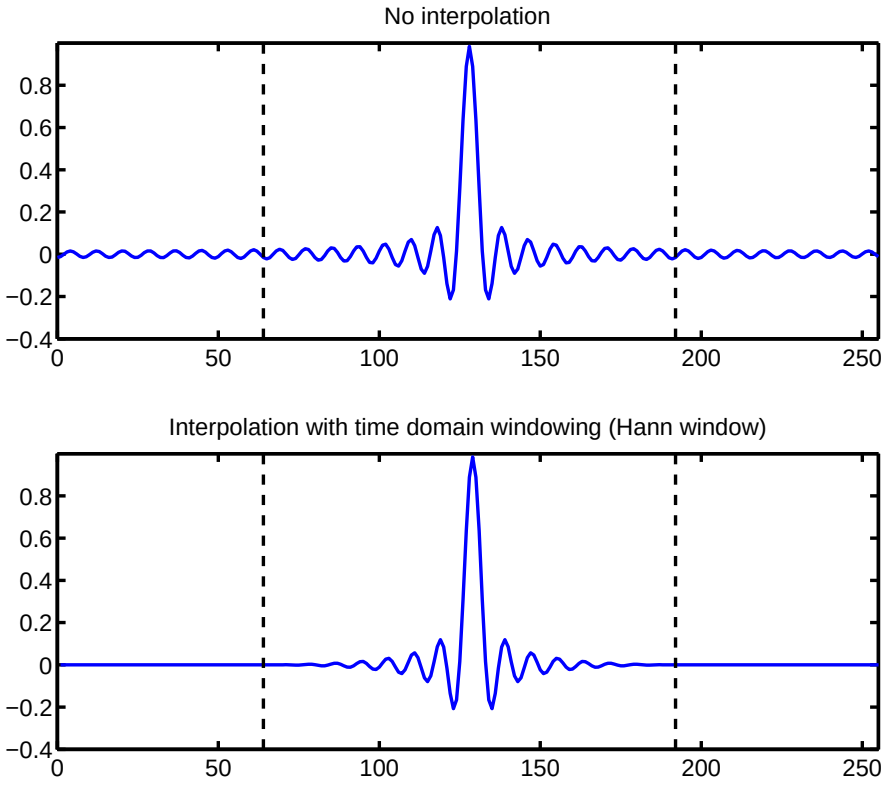


Figure 7.1: Impulse responses of an ideal $1/16^{th}$-band lowpass filter. The upper one is the full length generated from the frequency response with IDFT. The lower one has been windowed with a Hann window of half the length of the DFT. Length of the DFT is 256 points.

Chapter 8

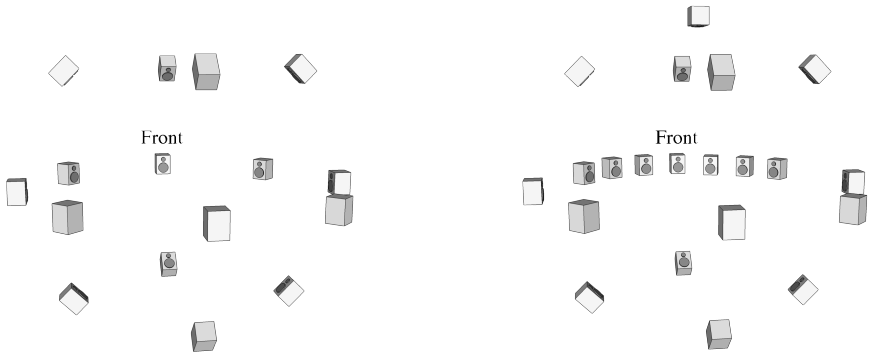
Results

This chapter describes the results found in informal evaluation performed by the author and the other members of the team developing Directional Audio Coding. Evaluation was performed in two environments. The first one was a medium sized anechoic chamber equipped with a three dimensional loudspeaker setup. For DirAC reproduction, a 16 loudspeaker symmetrical setup was used (see Fig. 8.1(a)). The reproduced sound was compared to an earlier high quality version of DirAC (Vilkamo, 2008) and a reference signal generated with virtual acoustics using a setup of 21 loudspeakers (see Fig. 8.1(b)). The second environment was a standard listening room (ITU-R BS.1116 standard) equipped with a setup of ten loudspeakers producing a half sphere of directions.

8.1 Multi-resolution STFT

The aim of using multiple concurrent STFT resolutions was to match the human hearing better and thus to reproduce challenging signals better. These challenging signals are generally impulsive signals like applauds and drum recordings. The general problem is that when there is enough frequency resolution for low frequencies to be reproduced correctly, time smearing will happen with impulsive sounds. However, using multiple resolutions essentially removes this time smearing and impulsive sounds do not degrade anymore.

Different resolution setups were informally tested and results were interesting. Using two resolutions with the frequencies under 1500 Hz using resolution of 1024 points and the high frequencies using resolution of 64 points produced a clear enhancement in the sound



(a) (b)

Figure 8.1: Loudspeaker setups used in the anechoic chamber. (a) A 16 loudspeaker symmetric setup for DirAC. (b) A 21 loudspeaker setup used for reference signals.

quality. Using three resolutions with the resolutions being 2048, 1024 and 32 points and with the band limits of 750 Hz and 3000 Hz, the difference compared to the two resolution version was still audible. However, it was nowhere as great as with the change from one resolution to two resolutions. Furthermore, a seven resolution "optimal" version spanning each power of two resolution from 2048 to 32 was tested. This version is based on the first method described in chapter 7.2.2. If this version listened to carefully, then it is possible hear some minor differences between this seven resolution version and the three resolution version. However, the increase in the required amount of computations is so large that it is not a good choice for real-time purposes.

In the end, two or three resolutions are the ones which should be preferred. However, these results should be verified with an actual listening test as the DirAC team might be highly biased and learned to listen to certain flaws in the algorithm.

8.2 Effects of bin-based processing

During the implementation, bin-based processing caused some expected and some unexpected effects. Firstly, calculating all information for each bin was predicted to require a lot of computation power. In synthesis this was the case but analysis was surprisingly light to calculate. Secondly, the amount of artifacts generated by double length synthesis filters was unexpected (although in hindsight it should have been seen) as previous STFT DirAC

implementations did not actually correct it but only applied double windowing[1] to remove the artifacts with some less important information. With bin-based processing the generated artifacts were intolerable and thus perfect interpolation had to be applied creating again more need for computation power.

The third effect is that the results in sound quality when compared to the frequency band processing are almost negligible when subjectively evaluated. For example the effects of chosen decorrelation scheme produce more audible differences than bin-based processing.

This all leads to a conclusion that at least for real-time purposes it is not feasible to use bin-based processing. Instead, equal subjective sound quality should be received by using frequency bands and applying double windowing or efficient polynomial interpolation. However, if theoretically more correct soundfield is desired, then it might be useful to apply bin-based processing.

8.2.1 Frequency smoothing

The effect of frequency smoothing was found to be a non-existent during the evaluation. It seemed that there was no effect when it was used in amounts which were presumed to be reasonable (that is with bandwidths under one ERB band). In a way this is logical as the human hearing itself applies averaging through frequency so if DirAC works at better resolution then it is only reasonable that there would not be any audible difference. However, it was a bit hoped for that by using frequency smoothing, some of the artifacts needing interpolation would disappear. Surprisingly, this was not the case and there was no effect whatsoever. However, it should be noted that the testing was done on a limited (although varying) set of sound samples and thus some other signals might produce different results.

8.3 Decorrelation methods

Different decorrelation methods presented in chapters 6.4.4 and 6.5.3 were evaluated for quality by listening for single decorrelated channels, subjectively listening for the feeling of diffuseness in the synthesized diffuse soundfield and listening to the complete result of DirAC synthesis compared to a reference sound. The artifacts generated by different algorithms were different so evaluation was bit hard to do.

[1] Double windowing means application of two window functions instead of one. The first one is applied before processing and the second one is applied after it.

The first evaluated method was the critical bands with random time shifts. This method was implemented by creating a filter for each channel with the described method. Frequency bands were divided in logarithmic fashion so that low frequencies had narrower bands than the high frequencies. Also, different number of bands were tested and best results were received when the number of bands was from 30 to 40. As for the amount of delay, a frequency dependent maximum delay was selected. At low frequencies the maximum of 40 ms was allowed and the maximum was linearly decreased to 20 ms for high frequencies. The minimum value was set to 5 ms for all frequencies. These values are related to the corresponding values of the precedence effect (see chapter 3.2.3). To enable the maximum delay, a 4096 point FIR filter was used. Resulting decorrelation was quite pleasant and with most signals, no other artifacts were heard except a slight change in timbre.

The second evaluated method was the decorrelation with noise bursts. The frequency dependent exponentially decaying noise bursts were selected. The filter length was again chosen to be 4096 points to enable better decorrelation at low frequencies. The resulting diffuse sound seemed good at first, but after a while, it was clear that the decorrelation was not strong enough and there seemed to be some timbre problems. These problems are probably caused by the fact that the low frequencies did not decorrelate well with the used filter length.

The third method was the hybrid decorrelation. It was straightforwardly formed from the two previous methods by applying two steep bandpass filters (1500 Hz being the band limit) and adding the results together. This combination filter with a length of 4096 points was applied to decorrelate channel signals. The result was that decorrelation seemed to be better than with the pure noise burst method. However, there still were some timbre problems.

The fourth and final method was the one based on time-varying phase shifts. The implemented method applied phase modulators at different frequencies for each channel. The maximum phase shift (and thus the maximum delay) was made frequency dependent so that on the high frequencies there was a smaller shift applied. This method was efficient to implement. The results were not that good as this method did not decorrelate signals enough. However, this might be the result of not choosing the parameters well enough as it was found that with a larger amount of loudspeakers the phase modulation could be much more drastic than presumed.

In the end, the best choice for decorrelation was the first one. This method was already used in the previous DirAC implementations and found to be the best choice also in them. However, other methods should be studied more as this method of decorrelation is still quite

heavy to calculate.

8.4 Efficiency

As DirAC is intended as a real-time audio processing tool, it is important that the algorithm is also efficient. Previous STFT implementations were already quite efficient and ran in real-time on a fairly new laptop computer [2]. However, new features implemented for this thesis created some more need for computation power.

Multi-resolution processing theoretically multiplies the number of needed computations if high-quality is desired. Additionally, bandpass filters have to be formed and applied for each resolution. However, in this implementation, a compromise was made and instead of correct bandpass filters, only the desired part of the bins were used in calculations. This would create some artifacts in the resulting sound but after already needed filter interpolation, they are not audible. Thus, the amount of needed computations is lessened significantly.

Bin-based processing also required a lot of computation power. Especially in synthesis this was a big problem as the synthesis filters are formed with a multitude of calculations. Additionally, the produced synthesis filters have to be interpolated with time domain windowing to remove all artifacts. Frequency smoothing also adds to the task as it is essentially convolution in frequency domain which is never a small task to do.

Decorrelation is also a quite heavy task to do as it applies an additional long FIR filter to each output channel after all other synthesis algorithms. There might a way of applying these filters during the other processing but with multiple resolutions it would require some creative programming.

This all resulted in the problem that the version implemented during this thesis requires a lot of computation power. On a fast general computer only a two resolution version ran in real-time. This was with the code heavily optimized and using vectorized computation. When the ideal case would be to run the same version with a normal computer using only a small amount of resources, this is hardly acceptable. Thus, it is imperative that a compromise is found between this theoretically high quality version and a fast version. This, however, is left for a future study with only a suggestion that frequency band processing is used instead of bin-based processing.

[2] Apple Powerbook G4 at the time

Chapter 9

DirAC software library

As a partial task for this thesis a software library was designed and produced to perform DirAC processing in multitude of situations. This chapter describes the properties of this software library in detail. First, a short look will be done to the general paradigm used and after that each relevant parameter will be described in more detail.

9.1 General design

The main purpose of the software library was to provide easily customizable tools for further scientific research and to offer a relatively easily portable package which can be distributed for interested parties. This was needed as previous implementations were designed around a single task and did not offer easy customizability. These reasons led to the use of standardized C as the programming language. It was a simple choice as C is a widely used programming language and even digital signal processors support it. Additionally, it optimizes well thus offering good real-time performance when desired. To enable vectorized calculation, the vDSP library of Mac OS X was used and thus the library is currently platform dependent. This, however, is hardly a restriction as the library was designed so that only a handful of simple functions need to be converted if another platform is used.

The library also includes some additional code to enable the use of DirAC in Max/MSP. This is because Max/MSP was the primary tool for testing the real-time functionality of the library.

The general design paradigm was based on creating functional blocks which perform inde-

pendently of each other. The signal is then moved between these blocks in a structure form containing a block of signal data. This block design enabled easy testing during development and offers a possibility to modify separate blocks without affecting the others. Also, access to signal data between blocks is easy and thus for example synthesis block can be used separately in other tasks.

9.1.1 Functional blocks

Like the DirAC algorithm, the software library is divided to functional blocks.

The first block is multi-resolution STFT which handles the conversion of a time domain signal to the multiple resolutions in frequency domain. To achieve this, the incoming signal is cut to smaller blocks in time fit for the current resolution. These blocks are then windowed and zero-padded to double length to ready them for later processing. After this the FFT is applied and the results are stored to a special signal structure.

The second block performs DirAC analysis. It is further divided to four separate functions which each perform their own task. These tasks are: calculation of intensity vectors, calculation of energy, estimation of direction of arrival and estimation of diffuseness. These functions essentially just apply the equations presented in chapter 6.3. Additionally, time and frequency averaging is performed in this block where needed. Division to frequency bands is also done here.

The next big block is the creation of virtual microphone signals. Its task is to produce correct virtual microphone signals from the input B-format signal using the equation in chapter 6.4.1. This task is optimized so that some of the values are pre-calculated to a table enabling faster processing.

Then comes the DirAC synthesis block. Actually this block is built from four blocks as direct sound synthesis, diffuse sound synthesis, decorrelation and VBAP act separately from each other. The actual synthesis parts simply apply the equations in chapters 6.4.2 and 6.4.3. VBAP is optimized by pre-calculating a table of values with a chosen resolution (one degree resolution being the default) and corresponding to the used loudspeaker setup. Time averaging is applied to the resulting VBAP gains if desired. Interpolation is applied to the synthesis filters to remove artifacts if this is desired. Synthesis block also converts the signals back to time domain and combines them with the overlap-add method. Second window function is also applied if it is desired. Decorrelation is then applied as a separate filter with the aid of FFT unless the time-varying phase shift method is used which is intertwined

inside diffuse synthesis algorithm.

Finally, a separate mixing block exists which adds direct and diffuse signals together to produce the final output signal. This is done with the aid of a ring buffer which offers optimal efficiency.

For portability purposes, a single code file implementing needed vectorized math functions has been created. In current implementation, it usually just wraps corresponding vDSP functions. By converting these functions to use corresponding math functions of the desired platform, it should be easy to use the library on different platforms than Mac OS X.

9.2 Parameters

This section describes different parameters which can be used to modify the performance of the DirAC algorithm in the library.

9.2.1 General

There are a few general parameters which affect the overall performance of the system. The sampling frequency and the input/output signal block size are self-explanatory. Only restriction is that the block size must be at least as large as the largest used FFT size. Additionally, there is a parameter for defining transmitted signals. This parameter defines which of the B-format signals will be transmitted from the analysis to the synthesis and thus affects the calculations. This parameter should be used to reduce the amount of transmitted data on low bandwidth applications.

9.2.2 Multi-resolution STFT

Multi-resolution STFT includes three parameters which are dependent on each other. The first one is the number of resolutions used. The two other then define for each resolution the actual resolution used and the frequency band where that resolution should be used. These parameters affect the quality of the processing greatly so it is important to choose them correctly.

Windowing

STFT also applies windowing to the signal. The chosen window for this implementation is the periodic Hann window. With a parameter, it is possible to select between single preprocess windowing or double windowing. With double windowing, the used window is a square root of the periodic Hann window (which is also known as MLT sine window).

9.2.3 Frequency bands

The use of frequency bands can be controlled with a set of parameters. The number of frequency bands and the division method can be selected. Also, the use of frequency bands can be turned off which results in bin-based processing.

9.2.4 Analysis

DirAC analysis is controlled by several parameters. First of all, the time averaging method is controlled by parameters defining the form of averaging and the amount of averaging. Currently, only one pole averaging is implemented but it relatively easy to extend this for other window types. The amount of averaging can be further controlled with a maximum value and a frequency dependency value. This should offer enough control for most purposes. If bin-based processing is used then frequency smoothing can be controlled by selecting the type and the relative bandwidth of the frequency dependent smoothing. Finally, the diffuseness estimation method can be selected from two choices implementing the two different algorithms.

9.2.5 Synthesis

DirAC synthesis is also controlled with several parameters. Time averaging of the loudspeaker gains can be similarly controlled as time averaging in analysis. Again, only one pole averaging is implemented. Virtual microphone type is controlled with a single parameter. Interpolation of synthesis filters can also be controlled with a parameter with the option being from no interpolation to perfect interpolation and different polynomial interpolations in between. The ground rule with this is to use as much as is needed to remove audible artifacts. The choice of decorrelation method is also a parameter and affects the sound quality quite a lot. Most decorrelation methods are implemented as simple FIR filters so this is a

choice between them. However, time-varying phase shift method has additional parameters defining the frequencies and amplitudes of the modulators which can be tuned.

9.3 Future additions

The software library is constructed so that future additions should be as painless as possible. A few additions are important to implement and certainly will done. One of them is the addition of head-tracking headphone reproduction which offers high quality headphone reproduction with relative ease. Also, some optimizations should be thought out as, for example, digital signal processors have more restrictive needs.

Chapter 10

Conclusions and Future Work

Directional Audio Coding (DirAC) has evolved from the original concept through a few iterative stages and versions. In this thesis a new version based on the short-time Fourier transform (STFT) was implemented. Additionally, a software library was developed for performing DirAC in multitude of situations.

The first half of this thesis described the background information needed for the theory of DirAC. Then the actual theory was presented with references to the discoveries done in the previous implementations. Also, new propositions were presented. Then, the done implementation was described and evaluated. Finally, a short documentation of the developed software library was presented.

The new propositions included a multi-resolution STFT scheme, bin-based processing including frequency smoothing and two decorrelation methods using a hybrid method and a time-varying phase shift method. The informal listening results showed that only multi-resolution STFT produced quality enhancement to the DirAC algorithm. Additionally, the amount of computation power required by the new additions leads to a conclusion that high computational accuracy is not justified. Instead, optimization should be done to the level where there is no significant quality issues when measured with listening tests.

In the end, the produced software library of DirAC can produce high quality spatial audio reproduction which was the original aim for this thesis work. Especially compared to the previous implementations, this implementation is at least on the same quality level and in some cases, much better.

As for the future work, it is highly recommended to perform formal listening tests to mea-

sure the effects of using multiple resolutions. Also, different decorrelation methods should be studied more as the current decorrelation is the one process requiring most calculation power in DirAC.

Bibliography

AES Staff. Wavefield synthesis: Evolution from stereophony and some practical challenges. *Journal of Audio Engineering Society*, 52(5):538–543, 2004.

Jukka Ahonen, Ville Pulkki, Fabian Kuech, Giovanni Del Galdo, Markus Kallinger, and Richard Schultz-Amling. Diffuseness estimation using temporal variation of intensity vectors. Manuscript, 2009.

Jens Blauert. *Spatial Hearing*. The MIT Press, Cambridge, MA, USA, revised edition, 1997.

Maurice Bouéri and Chris Kyriakakis. Audio signal decorrelation based on a critical band approach. In *Proceedings of the Audio Engineering Society 117th Convention*, 2004.

F. J. Fahy. *Sound Intensity*. Elsevier Science Publishers Ltd., Essex, England, 1989.

W. E. Feddersen, T. T. Sandel, D. C. Teas, and L. A. Jeffress. Localization of high frequency tones. *Journal of Acoustical Society of America*, 5:82–108, 1957.

Michael A. Gerzon. Ambisonics in multichannel broadcasting and video. *Journal of Audio Engineering Society*, 33(11):859–871, 1985.

B. R. Glasberg and Brian C. J. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47:103–138, 1990.

E. Bruce Goldstein. *Sensation and Perception*. Wadsworth, sixth edition, 2002.

David Griesinger. The psychoacoustics of apparent source width, spaciousness and envelopment in performance spaces. *Acta Acustica*, 83(4):721–731, July 1997.

E. R. Hafter. Spatial hearing and the duplex theory: How viable is the model? In G. M. Edelman, W. E. Gall, and W. M. Cowan, editors, *Dynamics aspects of neocortical function*. New York: Wiley, 1984.

M. O. J. Hawksford and N. Harris. Diffuse signal processing and acoustic source characterization for applications in synthetic loudspeaker arrays. In *Proceedings of the Audio Engineering Society 112th Convention*, 2002.

Jürgen Herre and Herbert Buchner. Audio signal decorrelator. International patent WO2007118583, 2007.

Matti Karjalainen. *Kommunikaatioakustiikka*. Helsinki University of Technology, Department of Signal Processing and Acoustics, P.O.Box 3000, FIN-02015 TKK, 2009.

Ole Kirkeby, Philip A. Nelson, and Hareo Hamada. The "stereo dipole" — a virtual source imaging system using two closely spaced loudspeakers. *Journal of Audio Engineering Society*, 46(5):387–395, May 1998.

Mikko-Ville Laitinen. Binaural reproduction for directional audio coding. Master's thesis, Helsinki University of Technology, May 2008.

R. S. Longhurst. *Geometrical and Physical Optics*. John Wiley and Sons Inc, 2nd edition, 1967.

Juha Merimaa and Ville Pulkki. Spatial impulse response rendering i: Analysis and synthesis. *Journal of Audio Engineering Society*, 53(12):1115–1127, December 2005.

Sanjit K. Mitra. *Digital Signal Processing – A Computer-Based Approach*. McGraw-Hill, 1221 Avenue of the Americas, New York, NY 10020, international edition edition, 2006.

Henrik Møller, Michael Friis Sørensen, Dorte Hammershøi, and Clemen Boje Jensen. Head-related transfer functions of human subjects. *Journal of Audio Engineering Society*, 43(5):300–321, May 1995.

Brian C. J. Moore, editor. *Hearing*. Academic Press, 1995a.

Brian C. J. Moore, editor. *Hearing*. Academic Press, 1995b.

H. Nélisse and J. Nicolas. Characterization of a diffuse field in a reverberant room. *Journal of Acoustical Society of America*, 101:3517–2524, 1997.

Mark Pinsky. *Introduction to Fourier Analysis and Wavelets*. Brooks/Cole, 2002.

Ville Pulkki. Virtual sound source positioning using vector base amplitude panning. *Journal of Audio Engineering Society*, 45(6):456–466, June 1997.

Ville Pulkki. Spatial sound reproduction with directional audio coding. *Journal of Audio Engineering Society*, 55(6):503–516, June 2007.

Ville Pulkki and Juha Merimaa. Spatial impulse response rendering ii: Reproduction of diffuse sound and listening tests. *Journal of Audio Engineering Society*, 54(1/2):3–20, January/February 2006.

Monique Radeau. Auditory-visual spatial interaction and modularity. *Current psychology of cognition*, 13(1):3–51, 1994.

L. Rayleigh. On our perception of sound direction. *Philosophical Magazine*, 13:214–232, 1907.

Thomas D. Rossing, F. Richard Moore, and Paul A. Wheeler. *The Science of Sound*. Addison Wesley, 3rd edition, 2002.

Julius O. Smith. Spectral audio signal processing (october 2008 draft). `http://ccrma.stanford.edu/~jos/sasp/`, October 2008a.

Julius O. Smith. Spectral audio signal processing (october 2008 draft), section: COLA examples. `http://ccrma.stanford.edu/~jos/sasp/COLA_Examples.html`, October 2008b.

Julius O. Smith. Spectral audio signal processing (october 2008 draft), section: Multiresolution STFT. `http://ccrma.stanford.edu/~jos/sasp/Multiresolution_STFT.html`, October 2008c.

Julius O. Smith. Spectral audio signal processing (october 2008 draft), section: Zero padding in the time domain. `http://ccrma.stanford.edu/~jos/sasp/Zero_Padding_Time_Domain.html`, October 2008d.

Juha Vilkamo. Spatial sound reproduction with frequency band processing of b-format audio signals. Master's thesis, Helsinki University of Technology, May 2008.

Juha Vilkamo, Tapio Lokki, and Ville Pulkki. Directional audio coding: Virtual microphone based synthesis and subjective evaluation. In review for the Journal of Audio Engineering Society, 2008.

W. A. Yost and G. Gourevitch, editors. *Directional hearing*, chapter The precedence effect. New York: Springer-Verlag, 1987.

E. Zwicker, G. Flottorp, and S. S. Stevens. Critical bandwidth in loudness summation. *Journal of Acoustical Society of America*, 29:548–557, 1957.